AD_____

Award Number: DAMD17-98-1-8061

TITLE: Application of Information Theory to Improve Computer-Aided Diagnosis

PRINCIPAL INVESTIGATOR: Doctor Paul Sajda
Doctor Clay Spence

CONTRACTING ORGANIZATION: Sarnoff Corporation
Princeton, New Jersey 08543-5300

REPORT DATE: August 2001

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 2001	Final (1 Jul 98 - 1 Jul 01)

4. TITLE AND SUBTITLE
Application of Information Theory to Improve Computer-Aided Diagnosis

5. FUNDING NUMBERS
DAMD17-98-1-8061

6. AUTHOR(S)
Doctor Paul Sajda
Doctor Clay Spence

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sarnoff Corporation
Princeton, New Jersey 08543-5300
email psajda@sarnoff.com

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Mammographic Computer-Aided Diagnosis (CAD) systems are an approach for low-cost double reading. Though results to date have been promising, current systems often suffer from unacceptably high false positive rates. Improved methods are needed for optimally setting the system parameters, particularly in the case of statistical models that are common elements of most CAD systems. In this research project we developed a framework for building hierarchical pattern recognizers for CAD based on information theoretic criteria, e.g., the minimum description length (MDL). As part of this framework, we developed a hierarchical image probability (HIP) model. HIP models are well-suited to information theoretic methods since they are generative. We developed architecture search algorithms based on information theory, and applied these to mammographic CAD. The resulting mass detection algorithm, for example, reduced the false positive rate of a CAD system by 30% with no loss of sensitivity. We showed that the criteria reliably correlate with performance on new data. The framework allows many other applications not possible with most pattern recognition algorithms, including rejection of novel examples that can't be reliably classified, synthesis of artificial images to investigate the structure learned by the model, and compression, which is as good as JPEG.

14. SUBJECT TERMS
Computer-Aided Diagnosis, CAD, Mammography, Model Selection, Minimum Description Length Principle, Hierarchical Image Probability, HIP, Hierarchical Pyramid Neural Network, HPNN

15. NUMBER OF PAGES
82

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the U.S. Army.

_____ Where copyrighted material is quoted, permission has been obtained to use such material.

_____ Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

X
_____ Citations of commercial organizations and trade names in this report do not constitute an official Department of the Army endorsement or approval of the products or services of these organizations.

_____ In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and Use of Laboratory Animals of the Institute of Laboratory Animal Resources, National Research Council (NIH Publication No. 86-23, Revised 1985).

_____ For the protection of human subjects, the investigator(s) have adhered to policies of applicable Federal Law 45 CFR 46.

_____ In conducting research utilizing recombinant DNA, the investigator(s) adhered to NIH Guidelines for Research Involving Recombinant DNA Molecules.

_____ In the conduct of research involving hazardous organisms, the investigator(s) adhered to the CDC-NIH Guide for Biosafety in Microbiological and Biomedical Laboratories.

7/26/01

Applications of Information Theory to Improve Computer-Aided Diagnosis Systems

Final Report

Clay Spence and Lucas Parra*
Sarnoff Corporation
CN5300
Princeton NJ 08543-5300
{cspence, lparra}@sarnoff.com

I. Introduction

Computer-aided diagnosis (CAD) systems for mammography, under development for more than 10 years, are an approach for low-cost double-reading with potential to improve the detection of breast cancer. Though results to date have been promising, current systems often suffer from unacceptably high false positive rates and lower than expected sensitivity and specificity when evaluated on new data. Improved methods are needed for optimally setting the system parameters, particularly in the case of statistical models and neural networks which are common elements of most CAD systems. This research project looks to apply principles from information theory to build improved statistical models for CAD systems.

Specifically, we develop a framework for building hierarchical pattern recognizers based on information theoretic criteria. The best-known example of such criteria is the minimum description length principal (*MDL*) pioneered by Rissanen [9]. Using these criteria we have developed a framework for building generative hierarchical image probability (HIP) models. Since the HIP framework is a generative model, i.e., it directly models the probability of the image given the image class; it is well-suited to compression and thus application of MDL. Along with conventional MDL we have evaluated predictive MDL (*pMDL*) and Akaiki's information Criterion (*AIC*). Although these are used to select the complexity of the model, we have also applied information theoretic criteria to select the wavelet basis on which these models are built. We applied these techniques to the problems of microcalcification and mass detection. We evaluated these techniques using mammographic mass and microcalcification datasets from The University of Chicago (UofC) and in all cases performance has been evaluated relative to the UofC CAD system [24], i.e., the HIP model augments the UofC CAD system.

We also rigorously evaluated the generative properties of the model for image synthesis and novelty detection. Analysis of the HIP model for synthesizing new mammographic images is important for understanding how the model captures image structure specific to mammographic masses. Novelty detection is particularly relevant since it would enables our system to establish confidence measures on detection, something which most current CAD systems do not offer. Finally we briefly tested the models that we trained for classification on the rather different problem of compression, to demonstrate the flexibility of this approach.

II. Body

The following are the three primary tasks under the first year of the project.

1. Apply and evaluate the utility of our hierarchical pyramid neural network (HPNN) architecture for improving mass detection in a CAD system.

---

* With help from Paul Sajda, Department of Biomedical Engineering, Columbia University, New York NY, 10027.

2. Develop basic MDL framework within context of building models for CAD applications–development of HIP framework.

3. Apply MDL framework to select optimal number of nodes (labels) for statistical (HIP) models.

The following are the two primary tasks completed under the second year of this project:

1. Further develop and evaluate the hierarchical image probability model, specifically focusing on the generative aspects of its architecture.
2. Apply and evaluate MDL framework for selecting architecture of hierarchical model. Compare MDL framework with other model selection methods.

The primary tasks of year three were as follows:

1. Apply MDL selection framework to select wavelet packet bases from wavelet libraries, and train HIP models with the resulting representation.
2. Examine ROIs and correlate ability to detect certain physical structure with information constraint learned by model.
3. Perform ROC analysis and qualitative inspection of ROIs to determine improvement in models' performance. Analyze on UofC clinical dataset to determine if generalization performance of new data has improved.

In the following sections we describe our progress in accomplishing these tasks. We refer to our year 1 report [25] or the papers included in the Appendix for a detailed description of the HIP model.

A. MDL and AIC

We begin with a discussion of information theoretic criteria for selecting between alternative models. There are at least two such criteria: the Minimum Description Length (*MDL*) criterion and the Akaike's Information Criterion (*AIC*). We have investigated the usefulness of these criteria for choosing Hierarchical Image Probability (*HIP*) models for classifying mammographic mass and microcalcification Regions Of Interest (*ROIs*). A typical result is shown in Figure 1. Both MDL and AIC track test A_z performance—MDL and AIC cost decrease as A_z performance on the test set increases. In the following we describe the two criteria and then suggest a methods to further improve the information theoretic selection criteria.

1. MDL

The minimum description length of a set of data is the length of the data encoded according to some probability model, which is the model we are trying to fit to the data, plus the length of the description of the model (Rissanen, 1983; Rissanen, 1996). The length of the encoding of the data is the negative log probability density of the data according to the model, plus a constant representing the precision with which the data must be specified. We ignore this constant when doing model selection, since it is the same for all models.

The code length of the model has two components, a term for coding the architecture, and a term for encoding the parameters. Suppose we are comparing models with different structures. For example, we may be comparing mixture density models with different numbers of mixture components. We will call the different models *architectures*. In this example, the number of mixture components needs to be encoded, and in general the specific architecture must be encoded. In practice this is often ignored, since it is a small contribution to the total description length.

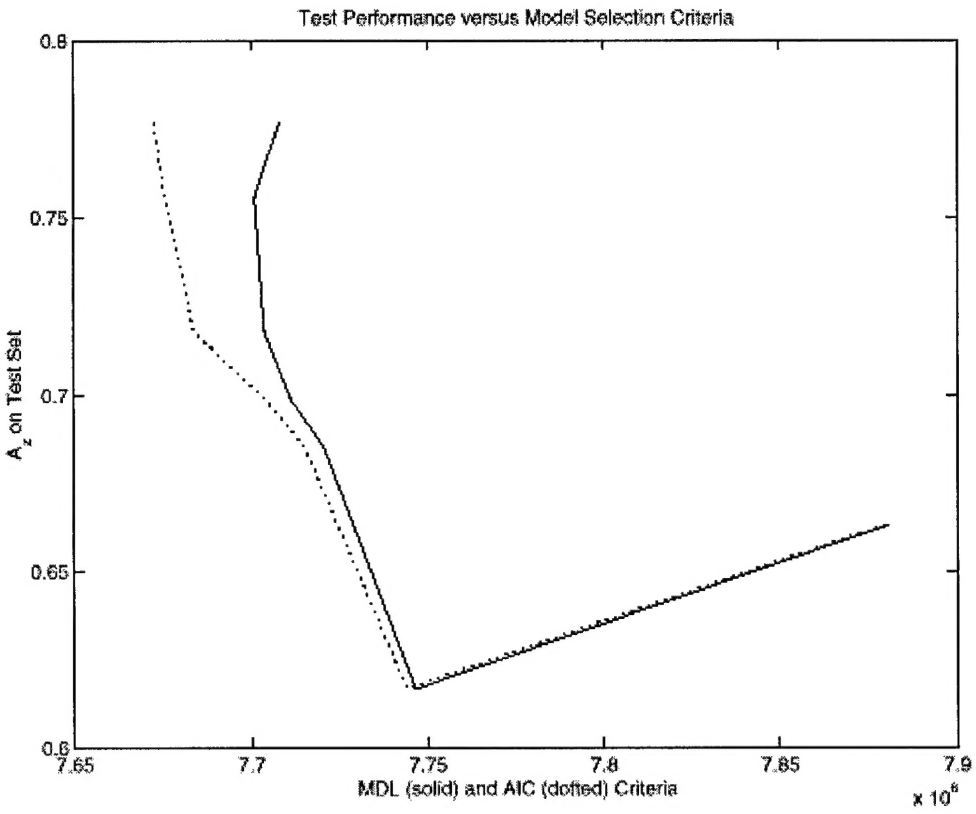

Figure 1. Information theoretic model selection using AIC (red) and MDL (blue). Plotted is model cost vs. A_z on the test data. MDL would choose a model with test $A_z = 0.75$ while AIC would choose a model with A_z=0.78. (The uncertainty in these estimates is probably greater than the difference.)

Given an architecture, we need to encode the parameter. The Cramer-Rao bound gives a lower limit on the variance of the parameters about their true value, assuming that the true probability is equal to our architecture with some values for the parameters. This limit is the inverse $\mathbf{M}^{-1}$ of the Fisher information matrix **M**,which is the negative expected value of the second derivative (or Hessian) with respect to the parameters of the log probability of the data according to the model, evaluated at the true value of the parameters. The precision with which we encode the parameters need not be greater than the precision with which we know them, i.e., it need not be greater than the standard deviations given by

$\mathbf{M}^{-1}$. Thus we would compute the components of the parameter vector along the eigenvectors of $\mathbf{M}^{-1}$, and the precision of these components are given by the square roots of the eigenvalues. The total code length of the parameters is the sum of the logarithms of these precisions, which is the log of the square root of the determinant of $\mathbf{M}^{-1}$. The code length for the parameters θ is the negative log of the square root of this volume, or

$$-\log(\left|\mathbf{M}^{-1}\right|^{1/2}) = \frac{1}{2}\log(\left|\mathbf{M}\right|). \tag{1}$$

Since it involves the probability of all of the data, **M** is proportional to the number of examples *N*, at least when there are enough examples. Because of this we can pull out the dependence on *N*. If there are *d* parameters, this gives

$$\frac{1}{2}\log(|\mathbf{M}|) = \frac{1}{2}\log\left(\left|\frac{N\mathbf{M}}{N}\right|\right) = \frac{1}{2}\log\left(N^d\left|\frac{\mathbf{M}}{N}\right|\right)$$
$$= \frac{d}{2}\log(N) + \frac{1}{2}\log\left(\left|\frac{\mathbf{M}}{N}\right|\right) \tag{2}$$

The second term is constant in the limit of large N, so in that limit we can ignore it. The remaining term is straighforward to compute, since we only need to know the number of parameters and the number of training examples. The total code length for MDL is therefore,

$$\text{MDL} = -\sum_{i=1}^{N}\log P(x_i \mid \theta) + \frac{d}{2}\log N \tag{3}$$

a. *Predictive MDL (PMDL)*

As stated earlier, minimum description length techniques lend themselves well to HIP models because a description length of the images given the HIP model naturally encodes the compactness of the HIP distribution along with the likelihood of the data under the HIP distribution. MDL therefore gives us a natural means for making various architecture choices, e.g., the number of labels at each level in the hierarchy, the types of features to use, and so on. In our first experiments we chose to use a *predictive MDL* or *PMDL* approach, due to its apparent simplicity. We take a training set S, say all of the mass ROIs in the complete training set, and give the images within it some ordering. We then train the HIP model on the first images in S and test it on some of the succeeding images. The test results in a log-likelihood for these test images, which we then use to initialize a running sum of test log-likelihoods. We then re-train on the first images plus the images on which we already tested, and test on more of the images, again adding the test result to the running sum of log-likelihoods. We repeat this until we have tested on the last images in S.

It has been shown [10] that the result is asymptotically equal to the description length of the model and the data under the final trained model. Intuitively, one expects a U-shaped curve as a function of model complexity. A model that is too simple will give relatively poor results late in the PMDL procedure, since it can't adequately fit the data. A model that is too complex will give relatively poor results early in the PMDL procedure, since it over fits the data, and in fact overfits it for more iterations than a simpler model. Thus models that are either too simple or too complex will have relatively high values for the accumulated test error. This intuition does not, of course, guarantee that the optimal model according to PMDL will generalize optimally, given the training data.

Compared to leave-one-out cross-validation PMDL should be quite fast because it starts each re-training run at an architecture that was optimized on a large fraction of the new training set. Besides the speed advantage, Rissanen claims that PMDL is more reliable than cross-validation [10]. We applied PMDL to choosing the number of components or labels at each level in HIP models for positive and negative ROIs, and we give those results in the experimental section.

Though PMDL seems simple, in fact for models of probability distributions ordinary MDL is straightforward, so we used plain MDL later in this project.

2. AIC

Akaike's Information Criterion is the expected Kullback-Leibler distance between the true model and the best model of the current architecture, given the data set. It is assumed that the architectures form nested sets with the true distribution being a member of one of these sets, that the number of examples N is sufficiently large, and that the current model is not too far from the true distribution. The resulting criterion is

$$\text{AIC} = -2\sum_{i=1}^{N}\log P(x_i \mid \theta) + 2d \tag{4}$$

3. Deficiencies of the information criteria

Both MDL and AIC assume that there are sufficent examples, *N*. Treatments of MDL, for example, sometimes use the term "asymptotically", which implies the number of examples goes to infinity *for a fixed model* (Rissanen, 1996). Thus we should only expect to get good results from these criteria if we have enough examples and we find a best model before trying models that are too large for the amount of data. In our current experiments we are not obviously in this situation. We have a fixed number of examples, and we are varying the model complexity. There is no criterion for deciding whether we have enough examples, or, alternately, when we have too complex a model for the criteria to be valid.

One possible method to address this "asymptotic" issue is to add corrections to the criteria For MDL, the correction is clear: include the second term from Equation (2). This is certainly more complex, but it is feasible. For the HIP model it should be possible to estimate the Hessian numerically.

B. HPNN

Prior to this project [11] we had developed a coarse-to-fine hierarchical pyramid/neural network *(HPN*N) architecture that combines multi-scale image processing techniques with neural networks to detect microcalcifications in digital/digitized mammograms (see Figure 2A). To *search* an image we apply the network at a position and use its output as an estimate of the probability that a microcalcification is present. We then repeat this at each position in the image. In the coarse-to-fine HPNN, the hidden units of networks operating at low resolution or coarse scale learn associated *context* information, since the targets themselves are difficult to detect at low resolution. The context is then passed to networks searching at higher resolution. The use of context can significantly improve detection performance since microcalcifications have few distinguishing features. In the HPNN, each of the networks receives information directly from only a small part of several feature images and so the networks can be relatively simple. The network at the highest resolution integrates the contextual information learned at coarser resolutions to detect the object of interest.

Under this project, we have extended the HPNN architecture by inverting the information flow in the coarse-to-fine architecture. This fine-to-coarse HPNN has networks extracting detail structure at fine resolutions of the image and then passing this detail information to networks operating at coarser scales (see Figure 2B). This is useful for many types of objects, such as mammographic masses, for which information about the fine structure is important for discriminating between different classes. Radiologists often distinguish malignant from benign masses based on the detailed shape of the mass border and the presence of spicules alone the border. Thus the fine-to-coarse HPNN should be well suited to this problem.

At each level of the fine-to-coarse HPNN several hidden units process the feature images. The outputs of each unit at all of the positions in an image make up a new feature image. This is reduced in resolution by the usual pyramid blur-and-subsample operation to make an input feature image for the network units at the next lower resolution. We trained the entire fine-to-coarse HPNN as one network instead of training a network for each level, one level at a time.

This training is quite straightforward. Back-propagating error through the network units is the same as in conventional networks. We must also back-propagate through the pyramid reduction operation, but this is linear and therefore quite simple. In addition we use the same UOP error function used in our previous work to train the coarse-to-fine architecture [12]. The rationale for this application of the UOP error function is that the truth data specifies the location of the center of the mass at the highest resolution. However, because of the sub-sampling the center cannot be unambiguously assigned to a particular pixel at low resolution.

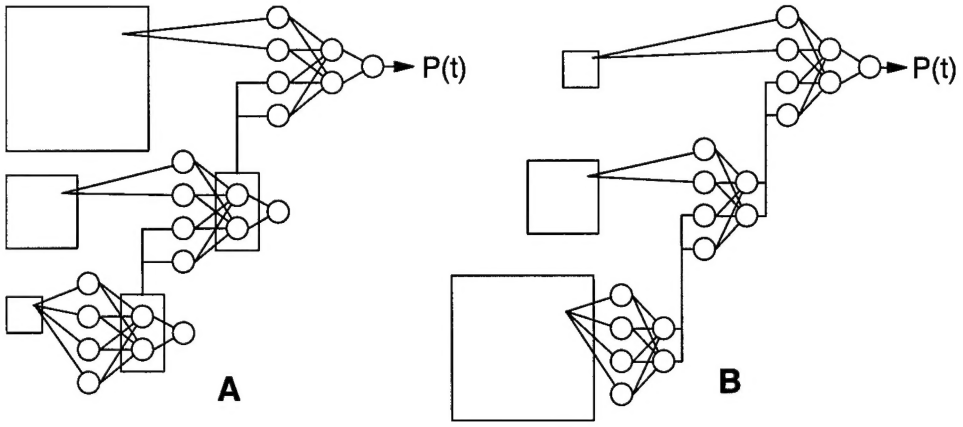

Figure 2. HPNN architectures. (A) The coarse-to-fine HPNN architecture exploits large-scale context to help detect small objects at fine scales. (B) The fine-to-coarse HPNN integrates fine-scale details to detect extended objects.

The features input to the fine-to-coarse HPNN are filtered versions of the image, with filter kernels given in polar coordinates by

$$\psi_{q,p}(r,\theta) = \left(\frac{q!}{\pi(q+|p|)!} \right)^{1/2} r^{|p|} e^{-r^2/2} L_q^{|p|}(r^2) e^{ip\theta} \tag{5}$$

where $L_q^{|p|}$ is an associated Laguerre polynomial. These have several convenient features. First, they are complete, so any image structure can be described in terms of them. Second, they are combinations of derivatives of Gaussians, and can be written as combinations of separable filter kernels (products of purely horizontal and vertical filters), so they can be computed at relatively low cost. Third, they are easy to steer, since this is just multiplication by a complex phase factor. We steered these in the radial and tangential directions relative to the tentative mass centers, and used the real and imaginary parts and their squares and products as features. The center coordinates of the tentative masses are generated by the earlier stages of the CAD system. These features were extracted at each level of the Gaussian pyramid representation of the mass ROI, and used as inputs only to the network units at the same level.

The fine-to-coarse HPNN is quite similar to the convolution network proposed by Le Cun, et al [5], however with a few notable differences. The fine-to-coarse HPNN receives as inputs preset features extracted from the image (in this case radial and tangential gradients) at each resolution, compared to the convolution network, whose inputs are the original pixel values at the highest resolution. Secondly, in the fine-to-coarse HPNN, the inputs to a hidden unit at a particular position are the pixel values at that position in each of the feature images, one pixel value per feature image. Thus the HPNN's hidden units do not learn linear filters, except as linear combinations of the filters used to form the features. Finally the fine-to-coarse HPNN is trained using the UOP error function, which is not used in the Le Cun network.

1. HPNN Mass detection results

As for microcalcifications [11], we apply the HPNN as a post-processor, but here it processes the output of the mass-detection component of UofC CAD system. The data in our study consists of 72 positive and 100 negative ROIs. These are 256-by-256 pixels and are sampled at 200-micron resolution. Half the data was used for training and half for testing.

Currently our best performing fine-to-coarse HPNN system for mass detection has two hidden units per pyramid level. This gives an ROC area (A_z) of 0.85 and eliminates 32 % of the false-positives without any loss in sensitivity. When tested on a third set of ROIs (36 positives and 200 negatives), the HPNN actually gave better performance, with Az increasing to 0.89 and eliminating 51 % of the false positives (Figure 3).

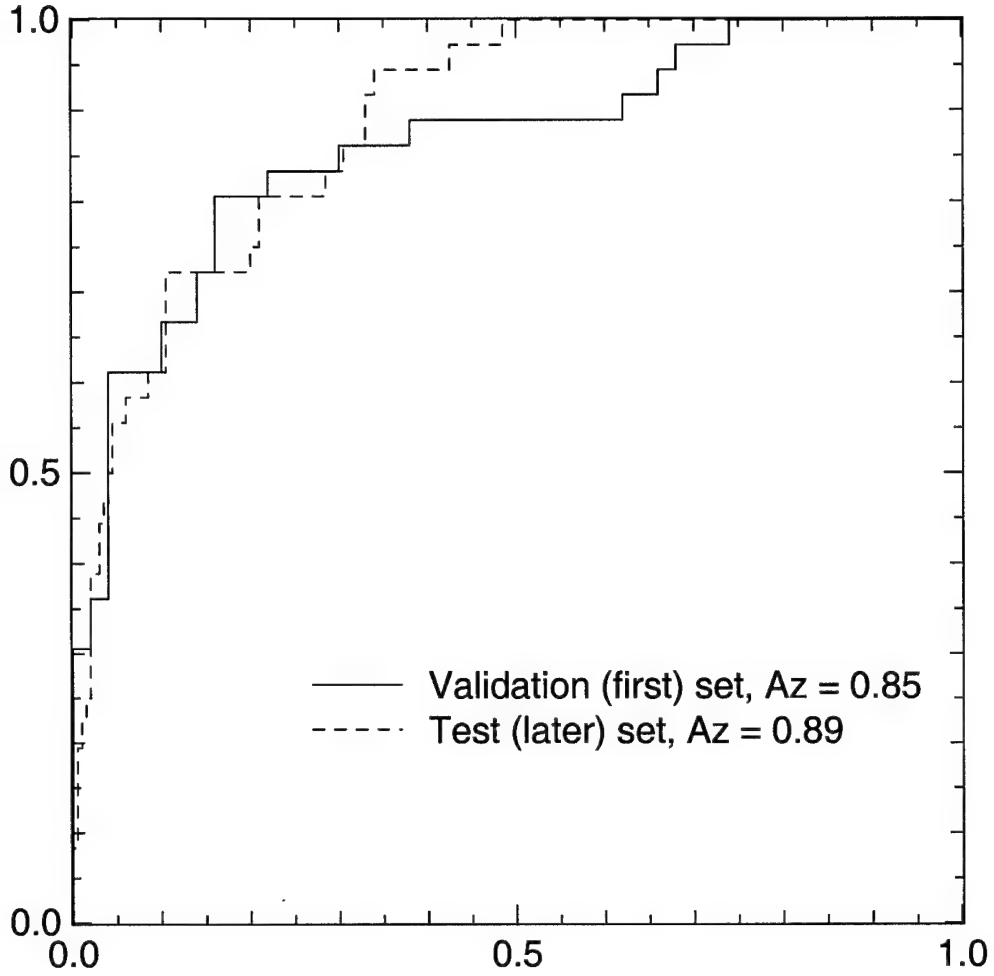

Figure 3. ROC curves of HPNN mass detector for validation set (part of the set from which training data was drawn) and a separate test set. (Data was provided by Prof. Maryellen Giger of the University of Chicago.)

C. HIP

Though our results for the application of the HPNN to mammographic CAD have been promising, the HPNN is not a good framework for applying MDL techniques for model selection. Since the HPNN estimates the class probability there is one bit of information per example (i.e. the ROI either has a mass or it does not). From the MDL point of view we are attempting to reduce the number of bits needed to specify the classes of a set of images by using a model to estimate those bits from the images. Since we need bits to specify the model and only one bit per image to specify its class without compression, we would need very many images to save more bits than those needed to encode the model. Most approaches to object recognition in images also estimate P(Class | Image), and so do not work well with MDL techniques.

Alternatively, if we estimate P(Image | Class), we are encoding an image, so we potentially save very many bits. In this case it is easy to save more bits compressing the image than it takes to specify the model, possibly even with a single image. This is important, given that many MDL techniques are only valid asymptotically [10], i.e., with a large amount of data. (A single image contains a large amount of data, in some sense, but the model structure determines whether this can be exploited.)

A model of the probability distribution of images has many other attractive features. We could use this for object recognition in the usual way by training a distribution for each object class and using Bayes' rule to get P(Class | Image). Since we would have P(Image), we could apply the resulting model to many different

tasks without further training or model selection. For example, we can detect unusual images and reject them rather than trust the classifier; something that is not possible with models of P(Class | Image). We could also use the mode to compress the images. Since a model of the distribution of data can be used to generate random examples that are supposedly typical of the data, this type of model is often referred to as a *generative* model.

Though the HPNN is not ideally suited for the application of MDL, it does have some attractive features. Most importantly, the HPNN is a framework for learning and integrating multi-resolution information for object classification. For instance, the HPNN is able to improve microcalcification detection performance for the University of Chicago CAD system because it can exploit low resolution contextual information, such as the location of blood vessels and the ductal system [11]. Thus a generative modeling framework should also take advantage of multi-resolution information for exploiting contextual information.

We have developed a model of image distributions with these properties, that we call the *Hierarchical Image Probability* or *HIP* model. In the following section we briefly describe previous work in modeling the probability distributions of images. We then describe the new framework of HIP models we have developed. We present the theory behind the framework and then our results in applying the HIP model to mammographic mass and microcalcification detection. We include discussions of several topics mentioned above, namely novelty detection, synthesis of images as a means of investigating the models, and compression. We also present our investigation of information theoretic techniques for choosing wavelet packet bases on which to build HIP models.

1. Theory

a. *Previous work in modeling image probability distributions*

Many image analysis algorithms use probability concepts, but few treat the distribution of images. Zhu, Wu and Mumford [14] do this by computing the maximum entropy distribution given a set of statistics for some features. This works well for textures but it is not clear how well it will model the appearance of more structured objects. In addition, with their approach it is easy to compute the probability of an image but harder to sample from the distribution, i.e., generate new artificial images. The ability to sample is necessary for many image analysis applications, e.g., compression.

There are several algorithms for modeling the distributions of features extracted from the image, instead of the image itself. The Markov Random Field *(MRF)* models are an example of this line of development; see, e.g., [6, 4]. Unfortunately they tend to be very expensive computationally. Because it is not an image distribution it only applies to some image analysis tasks, such as texture classification, that do not require sampling.

In De Bonet and Viola's flexible histogram approach [2, 1], features are extracted at multiple image scales, and the resulting feature vectors are treated as a set of independent samples drawn from a distribution. They then model this distribution of feature vectors with Parzen windows. The flexible histogram approach has given good results, but the feature vectors from neighboring pixels are treated as independent when in fact they share exactly the same components from lower-resolutions. To fix this one might build a model in which the features at one pixel of one pyramid level condition the features at each of several child pixels at the next higher-resolution pyramid level.

The multi-scale stochastic process *(MSP)* methods do exactly that. Luettgen and Willsky [8], for example, applied a scale-space auto-regression (AR) model to texture discrimination. They use a quadtree or quadtree-like organization of the pixels in an image pyramid, and model the features in the pyramid as a stochastic process from coarse-to-fine levels along the tree. The variables in the process are hidden, and the observations are sums of these hidden variables plus noise. However, the MSP model distributions are Gaussian, i.e., the joint distribution of all of the variables is a Gaussian distribution. This is clearly not the case in natural images, such as mammograms. Buccigrossi and Simoncelli [3], for example, have found that the distributions of some features have high kurtosis, and that the distribution of one feature conditioned on a neighboring feature has a "bow-tie" shape, which cannot follow from a Gaussian joint distribution. We have obtained similar results using our HIP model (Figure 4). The MSP approach also models the probability of the observations on the tree, not the probability of the image.

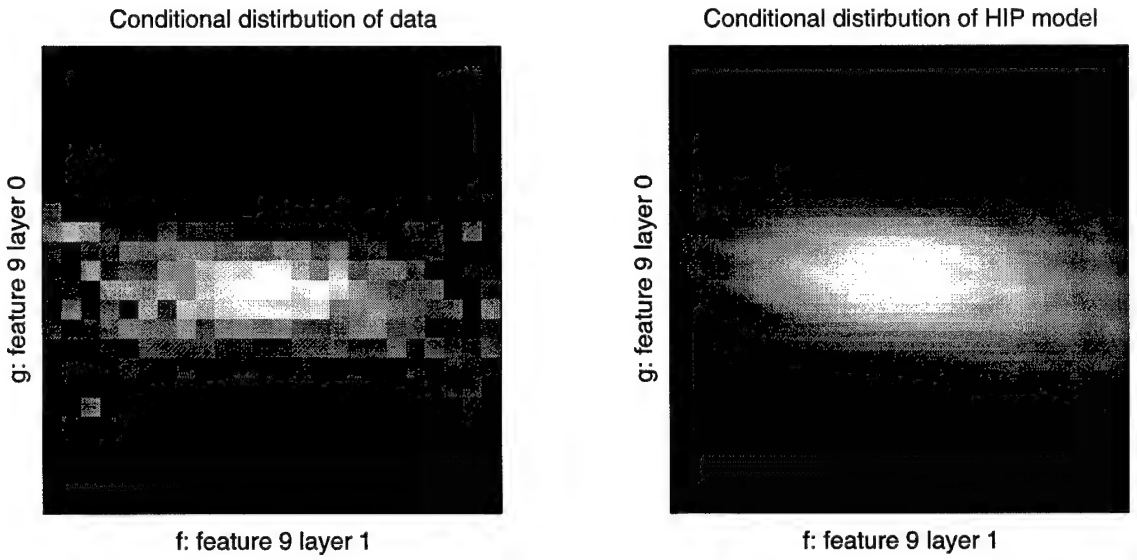

Figure 4. Empirical (left) and modeled (right) conditional histograms of image feature pairs.

All of these methods seem well-suited for modeling texture, but it is unclear how we might build the models to capture the appearance of more structured objects or objects which are hybrid in nature (e.g. which include both structure and texture), such as mammographic masses. We can argue that the presence of objects in images can make local conditioning like that of the flexible histogram and MSP approaches inappropriate. Objects in the world cause correlations and non-local dependencies in images across different resolutions. For example, the presence of a particular object might cause a certain kind of texture to be visible at some resolution. Usually the local image structure at lower resolutions by itself will not contain enough information to infer the object's presence, but the entire image at lower resolutions might. Therefore the probability that a texture is present will depend on a large region in the lower-resolution image.

Similarly, objects create long-range spatial dependencies at a given resolution. For example, an object class might result in a kind of texture across a large area of the image. If an object of this class were always present, we would know that the texture is present. But if such objects are not always present and cannot be inferred from lower-resolution information, knowing that the texture is present at one location tells us that it is present elsewhere.

These considerations imply that the assumptions of the flexible histogram and MSP approaches limit their capabilities. The features at one resolution and one location depend on lower-resolution image information over a large area of the image, and even given that information they depend on the features at other locations at that resolution.

More recently others have developed models similar to our HIP model. Crouse et al [15] developed a class of models they called Hidden Markov Trees (*HMTs*) that differ in various details from HIP models, but share much of the same spirit. They tend to emphasize the high kurtosis of the marginal distributions of wavelet coefficients, which they mode with a mixture of two Gaussians. Our HIP model is a little more general, at least in some ways, but they seem to have been quite successful in applying HMTs to several areas such as image denoising and segmentation. Also Cheng and Bouman [16] developed similar tree-structured image probability models for segmentation as extensions of Bouman and Schapiro's work on tree-structured multi-resolution segmentation [17].

b. HIP architecture and variations

Here we briefly describe HIP models and discuss variation in their architectures. A HIP model is built on the assumption that image information should be represented at various length scales and that it is usually good to condition fine scale information on coarser. For example, given an image of some object at some resolution, if we can identify the object we know a great deal about its likely appearance at higher

resolution. In these cases conditioning on low resolution information makes the higher resolution information less dependent at different locations. Accordingly we decompose the image into some multi-resolution representation such as a wavelet pyramid decomposition, like the of Simoncelli and Adelson [18]. Any invertible decomposition would do, since we are then just expressing the distribution in a different coordinate system. We then build a model of the probability distribution of each pyramid level in the decomposition, conditioned on the next coarser level, if there is one. These distributions are still very complex for most classes of images, so we would like to simplify them further by factoring into individual distributions over the wavelet coefficient vectors at each position. By itself this is too strong an assumption, but it is necessarily true that if we further condition on appropriate information then the distribution does factor, i.e., the coefficient vector at a position is conditionally independent of the vectors at other positions. The problem is finding the "appropriate information." Accordingly we add hidden variables to the model, build the model so that the model distributions of coefficient vectors are conditionally independent, and fit this to the data.

Our choice for hidden variables is strongly constrained by computational constraints. Accordingly we made two choices. First, the hidden variable at a position in a level conditions the wavelet coefficient vector at that position, and does so by indexing a multivariate Gaussian distribution for that vector. Thus we are building something like a Gaussian mixture model for the coefficient vectors. Dependence between levels is introduced by giving the means of the Gaussians a linear dependence on the parent wavelet coefficients.

Our second choice is to give a tree structure to the hidden variables. That is, at each position in a level the hidden variable depends only on a hidden variable at the next coarser level at the *parent* position. This position is determined by the subsampling operation of the wavelet decomposition. These parent-child relationships then determine a tree or quad-tree like structure. Since the hidden variables are indices of mixture components, they are essentially integers in some range. There is therefore a mixture model for the coefficient vector at a position, but the probabilities of the components are determined by the rest of the image. The same mixture components are used at all locations in a level, but we are free to use different components and different numbers of components at each level.

A minor but important elaboration is needed to model spatial patterns. Originally we assumed the distributions at each child position of a parent (upper and lower left, and upper and lower right, for example) were the same. We believe it is better to allow these to be different, so that a parent label can indicate particular spatial patterns, such as an edge passing through the upper two child positions. We have kept this in all except our earliest work.

That is a basic description of our earliest HIP models. We will often refer to the hidden indices as *labels*. These labels serve a dual purpose: first to determine a Gaussian mixture component, and second, to determine the hidden labels at finer resolutions. These two functions are somewhat at odds. For example, the hidden label at the very root of the tree may need to have a large number of possible values to cover different types of images within the class it is modeling. However there is only one wavelet coefficient vector per image at the root, and this vector may have several dimensions, requiring quite a few parameters per mixture component. The number of possible values of the hidden label is limited because we need sufficient examples *per label value* to fit the mixture component's parameters. We have addressed this by altering the model to have two hidden labels at each position. One hidden label, which we refer to as the *mixture label*, serves as the index of mixture component and depends only on the other hidden label. The second hidden label, which we call the *hierarchy label*, conditions only the local mixture label and the child hierarchy labels. This makes it possible to have few mixture components and many hierarchy labels at low-resolution pyramid levels.

We have also divided the mixture labels into what might be called a pattern label, though we continue to refer to it as a mixture label, and a scale label. This is intended to add some of the structure that Wainwright and Simoncelli found useful in modeling wavelet coefficient statistics [19]. They found that much of the statistics between pairs of wavelet coefficients at different positions, orientations, and/or scales can be fit by a model with Gaussian distributions for the coefficients and a hidden scale at each position, conditioned in a tree structure like the hidden labels in HIP. Their scale variables are continuous, however. In our case the pattern label indicates the mean, covariance matrix and correlation matrix that should be

used, up to a scale. The scale label indicates a factor by which the mean, covariance and parent correlation will be scaled.

There are many possible variations on these divisions of the hidden labels into different parts. All of them basically add substructure to the labels and their dependencies. A version we have recently used has the mixture, scale and hierarchy labels. The mixture and scale labels only condition the Gaussian distribution of the local coefficient vector, as described above. The hierarchy label conditions the mixture, scale and hierarchy coefficients *at the child positions*, not at the same level. In this way somewhat different functions are separated: the local image structure at a given scale and position is influenced by one pair of hidden variables, while a separate hidden variable is used to influence the spatial arrangement of such structures at finer resolutions.

c. *Architecture Search*

Information theory gives us criteria by which to judge models, but this alone does not tell us how to select models to compare, i.e., what path through the space of possible models we should take to find the best model. For the HIP model the search is over vectors of natural numbers, so there is no gradient to follow to find even a local minimum.

Initially we used PMDL as the search criterion along with a pseudo-gradient descent algorithm. In this algorithm we computed how the PMDL cost would change when the number of labels at one level was changed. We then searched along the vector over levels of these changes for a minimum. This search algorithm is sub-optimal. It can get stuck in local minima, and in fact one might say it can get stuck at many points that are not even local minima, since the points at which the algorithm exits are only better than those neighbors that differ in one component of the architecture vector. If changes in more than one component are allowed, many of these final points are probably not local minima. Unfortunately, the number of neighbors that differ in more than one component is very large, and since training one architecture already takes several hours on a Sun Ultra 60 workstation, this more complete search takes a prohibitively long time.

Furthermore we frequently find architectures and layers for which increasing and decreasing the number of components both give a decrease in the cost, yet we only search in one of the two directions, thus probably missing better local minima part of the time.

One alternative to these heuristic approaches is exhaustive search, at least in a bounded region of the search space. This is optimal but very expensive. Unfortunately the unknown behavior of means there are no better guaranteed-optimal methods.

It may be possible to develop a split-and-merge algorithm like that of Ueda, et al [13]. Such an algorithm would analyze the data conditioned on the model parameters, and attempt to decide which labels in the model could be merged and which could be split. In this way a new architecture is always initialized at a relatively good fit to the data, rather than with random starting values. This precludes using PMDL to judge whether a particular split-and-merge operation improved the architecture, so we would have to use a conceptually more straightforward estimate of code length. We have spent a little time investigating split and merge criteria, but with no firm conclusions yet.

In much of our work we tried intuitive approaches. In one such approach we used a HIP architecture in which the mixture labels only condition the Gaussians, i.e., the hierarchy labels transmit all of the information between levels. We began with one hierarchy label per level, so that they passed no information. In effect the HIP model was a set of independent mixture models, one for each level. We began with one mixture component and successively split and retrained the components, stopping when the AIC or MDL cost began to increase, or when we decided not to spend more computer time. We then added hierarchy components, successively splitting these and retraining, stopping according to AIC or MDL. In effect we build as good a model of the distribution of coefficient vectors as the data allows, and then use the hierarchy components to learn spatial relations between the mixture components. This tended to result in very many mixture components and few hierarchy components. In effect it spends most of the parameters modeling the marginal distributions of the coefficient vectors.

In a second approach, we first only split hierarchy labels (with a small number of mixture labels), effectively spending parameters on spatial relationships between the mixture components. A third approach is to alternate between splitting the mixture and hierarchy labels. All of these seem to be workable, though the approach of splitting mixture labels first seems to suppress the usefulness of hierarchy labels. The biggest problem is lack of computer time. Though we have tried to optimize the HIP training programs for speed, these search algorithms result in very large models that consume a great deal of memory and training time. In fact we have not yet reached a minimum of the AIC cost.

In summary the problem of architecture search for HIP models remains open. We have some heuristic approaches using information theoretic cost criteria that may be adequate. Our chief difficulties are the computational resources needed for the large HIP models.

d. Choice of image representation

In our early investigations of image synthesis with HIP models we noticed that our filter sets tended to suppress high frequencies. This means that the inverse transformation (reconstructing an image from the filtered and sub-sampled images) must boost these frequencies. In extreme cases this will result in ringing. In less severe cases there is a tendency toward "blockiness". This appears if we generate a set of white noise images, and then construct the original image that would have given these white noise images as feature images. That is, we assume the white noise images are feature images and reconstruct the corresponding original image. With our previous features this tended to give sharp horizontal and vertical edges that nearly group into squares.

The HIP model can partially eliminate this blockiness because it captures correlations between features at neighboring levels. Ignoring these correlations gives increased blockiness. While it is good that the HIP model can learn to eliminate artifacts such as blockiness, it is not a good use of the HIP model's resources since these artifacts are introduced because of the choice of features. We would prefer to have features that do not have such artifacts, so the resources of the HIP model can be devoted to learning other structures.

We studied this problem by trying to choose better sets of features or wavelets. We have designed even-tap wavelets for subsampling by two, and odd-tap wavelets for subsampling by three. (Curiously, when subsampling by two the resulting wavelets are only approximately orthonormal, but for subsampling by three there are exactly orthonormal wavelets.) We concluded that these minimize the tendency for blockiness, but splitting scales into discrete bands, i.e., pyramid levels, inevitably introduces this tendency, since a flat power spectrum at each level corresponds to a stepped power spectrum of the corresponding image. Probably we could further reduce blockiness through the use of overcomplete representations like Freeman and Adelson's steerable pyramids [20]. However HIP would no longer be a model of the image distribution, at least not obviously. Instead it is a model of the distribution of feature pyramids. For classification problems this may not matter.

2. Experiments

a. Mass Classification

We applied HIP to the problem of mass detection in mammographic CAD. We used the same data that was used to evaluate the HPNN (72 true positive and 100 false positive ROIs taken from the UofC CAD system). For this particular model $A_z = 0.79$. A comparison between the HIP and HPNN performance on the same data is shown in Table 1. Though the HIP model's performance on the test data was not as high as the HPNN, our efforts at model selection were limited by the long training time and high memory cost of the complex HIP models that perform better. Thus we hope that HIP models can perform as well given the same amount of data, but they are more costly to train. This is perhaps inevitable since estimating the image distribution is a harder task than estimating the conditional class probability.

Using this new hidden label architecture, the best HIP model pair, as defined by the AIC cost (see below), gives $A_z = 0.78$, and has the ROC curve shown in Figure 5. In this case we have eliminated 30% of the false positives of the UofC CAD system for mass detection, without loss in sensitivity.

Sensitivity	Fine-to Coarse HPNN Specificity	HIP Specificity
100%	51%	25%
95	57	36
90	67	52
80	79	75

Table 1. Specificity vs. sensitivity for the HPNN and HIP mass detectors.

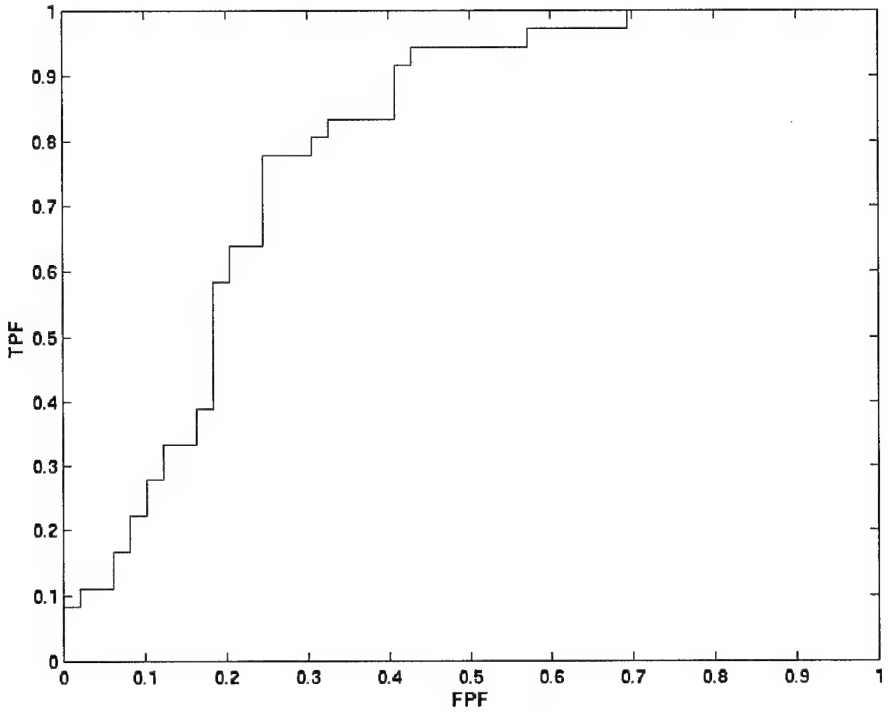

Figure 5. ROC curve of best HIP model, chosen using AIC. Results are relative to UofC CAD system for mass detection.

b. Novelty Detection

Novelty detection identifies examples that are significantly different from the examples on which the model(s) was trained [23]. Detecting novel examples can be useful in a CAD system for generating confidence measures on the CAD output and identifying data that could be used in future training of the neural network/statistical model. The HIP model's generative structure enables novel examples to be identified by thresholding the log-likelihood of the models. Figure 6 illustrates how ROC performance improves if novelty detection is used to generate a confidence measure for rejecting low-confidence examples. In this example, two HIP models were trained, one for positive ROIs and one for negatives ROIs (same ROI database as for classification and synthesis). Test data was evaluated by computing the likelihood ratio of the models as well as the absolute value of the log-likelihoods. The absolute values of

the log-likelihoods are thresholded such that low values are considered low confidence and therefore rejected (not classified). As the threshold on the log-likelihood is increased, more ROIs are rejected because of low confidence and the area under the ROC curve begins to increase. Also shown in Figure 6 are data that are rejected (not classified) because they fall below the threshold at the given rejection rate—these ROIs are novel with respect to the data on which the models were trained.

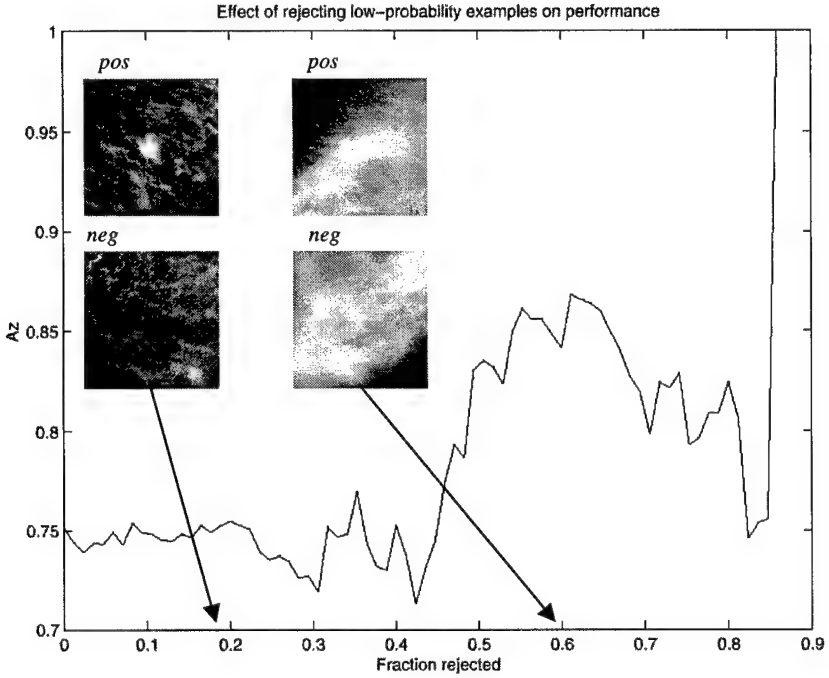

Figure 6: Novelty detection for improving ROC performance. The log-likelihood of the two HIP models (positive and negative) can be thresholded so that we reject (do not classify) a fraction of the test data that is novel, relative to the training examples. Shown is the area under the ROC curve as this novelty/confidence threshold is increased (thus increasing the fraction rejected). Also shown are examples of negative and positive ROIs that would be rejected at different thresholds.

c. Wavelet Bases

In the usual wavelet decomposition, basis filters are applied to the image and the results are subsampled, giving four images or subbands, each with one-fourth the number of pixels of the original image. This procedure is successively applied to the low-pass subbands to build the wavelet pyramid. Applying the decomposition to the low-pass subbands is a choice; it reflects the property of typical images that only the low-frequency components are correlated across long distances. In some images the higher frequencies can also be correlated over long distances. Ordered textures like fabrics are good examples of this. Saito has suggested a procedure for choosing a different wavelet decomposition to exploit such image structure [21].

If every subband is decomposed, the resulting set of subbands forms a tree with edges from each subband to those subbands that result from decomposition. The leaves of the tree are single-pixel images with very specific frequency content. There are as many leaves as pixels in the original image. The collection of such subbands is overcomplete; at any node in the tree the subband contains all the information in any set of its descendents. A complete set of subbands is any set that contains one and only one subband on any path from the root of the tree to a leaf. Saito called this a wavelet packet basis.

Saito suggested using an entropy measure to choose between these complete sets of subbands. This entropy is obtained by normalizing the sum of the squares of the pixels in the image before decomposing it. Because the wavelet transform is orthonormal, the sum of the squares of the pixels in any complete set of subbands will also be equal to one. Call the square of a pixel e (for energy), the entropy is defined as

$$H = -\sum_{b,x} e_{b,x} \log(e_{b,x}), \tag{6}$$

where the sum is over subbands b and positions x. This is a measure of compactness or sparseness. It is maximized if every pixel in every subband has the same energy, and minimized (equal to zero) if only one pixel in one subband has non-zero energy.

Given the tree structure of the wavelet decomposition, Saito searches each branch recursively, comparing the partial entropy of a node with the minimal energy of all complete sets of the node's children. This is tractable because the partial entropy of a node is independent of the partial entropy in other branches of the tree.

We have experimented with this approach to choosing a wavelet packet basis, using several image classes. It appears to be useful for highly structured images like some textures, as suggested above. Mammograms, on the other hand, seem to lack sufficient structure to make this as useful, but the technique has shown that we can usefully decompose all of the level-one subbands (level zero being the original image), giving a wavelet packet basis with relatively high dimensional level two coefficient vectors. The advantage is some memory savings in the HIP model, since the highest resolution level is then smaller, and the memory needed for HIP's internal variables during training is reduced.

d. Mass Synthesis

Since the HIP model is a generative model, we can sample the model and synthesize new images. In practice, this property might be best utilized for image compression or noise reduction. Within the context of ROI classification, synthesized images can give us insight into what features the model is extracting and representing for both positive and negative ROIs. Using the same ROI database used for classification, we constructed HIP models for positives (cancer) and negatives (no cancer). The trained HIP models were sampled to synthesize new ROI images. Figure 7 shows examples of these images. Inspection of the synthesized positive ROIs shows more focal structure, with more well-defined borders and higher spatial frequency content than the negative ROIs.

The sampling procedure begins at the coarsest resolution, where the hidden labels are randomly sampled from the distribution $P(A_L)$. The feature (wavelet coefficient) images $\mathbf{G}_L$ are then sampled from $P(\mathbf{G}_L \mid A_L)$. The $\mathbf{G}_L$ are used to construct I_{L-1}, from which the parent feature (wavelet coefficient) images $\mathbf{F}_L$ are constructed. We then sample A_{L-1} from $P(A_{L-1} \mid A_L)$, and then $\mathbf{G}_{L-1}$ from $P(\mathbf{G}_{L-1} \mid \mathbf{F}_L, A_{L-1})$. This is repeated until the finest resolution is reached and I_0 is constructed.

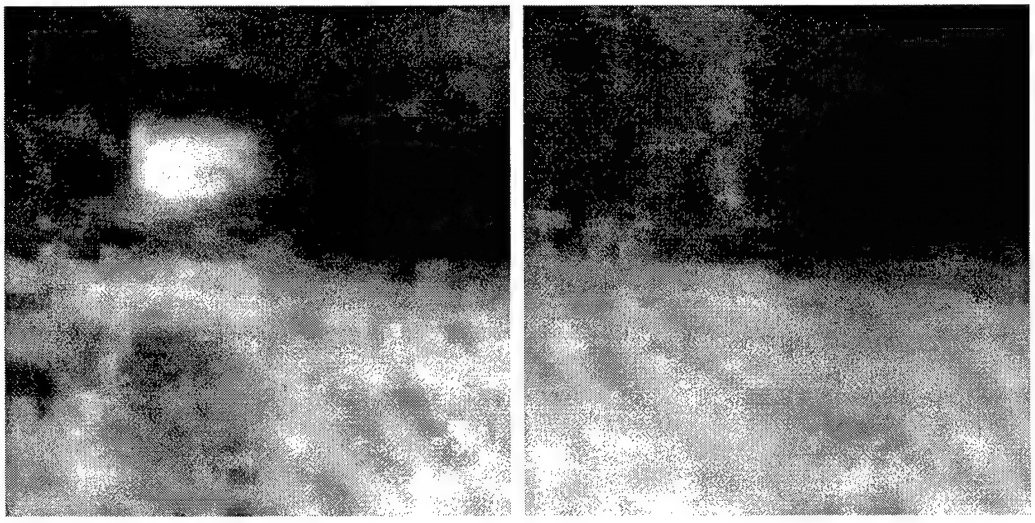

Figure 7: Mammographic ROI images synthesized from positive (left) and negative (right) HIP models. Synthesized positive ROIs tend to have more focal structure, with more defined borders and

higher spatial frequency content. Negative ROIs tend to be more amorphous with lower spatial frequency content.

e. Mass Compression

A stream of random variables can be optimally compressed if we know their distribution, and so having a HIP model of a source of images should allow us to compress examples of those images with high efficiency. Here we demonstrate compression with HIP models using a simple technique.

Given an image and a HIP model, we compute the most likely value of each hidden label,

$$a_l^*(x) = \underset{a_l(x)}{\text{argmax}}\, P(a_l, x, I)\,. \tag{7}$$

We then code each feature vector $\mathbf{g}_l(x)$ using $P(\mathbf{g}_l \,|\mathbf{f}_{l+1}, a_l^*, x)$. The latter is used by decomposing $\mathbf{g}_l(x)$ into its components along the eigenvectors of the covariance matrix of $P(\mathbf{g}_l \,|\mathbf{f}_{l+1}, a_l^*, x)$, $\Sigma_{a_l^*}$, and coding those components with a specified precision using Huffman encoders for the Gaussian distributions with variances given by the eigenvalues of $\Sigma_{a_l^*}$. The resulting bitstream was stored in a file that was susbequently compressed with gzip to reduce the redundancy in the many short identical bit patterns. This procedure is currently very computationally expensive, and is not necessarily optimal even if the HIP model exactly matches the image distribution, but it is straightforward to code and serves to demonstrate the capability.

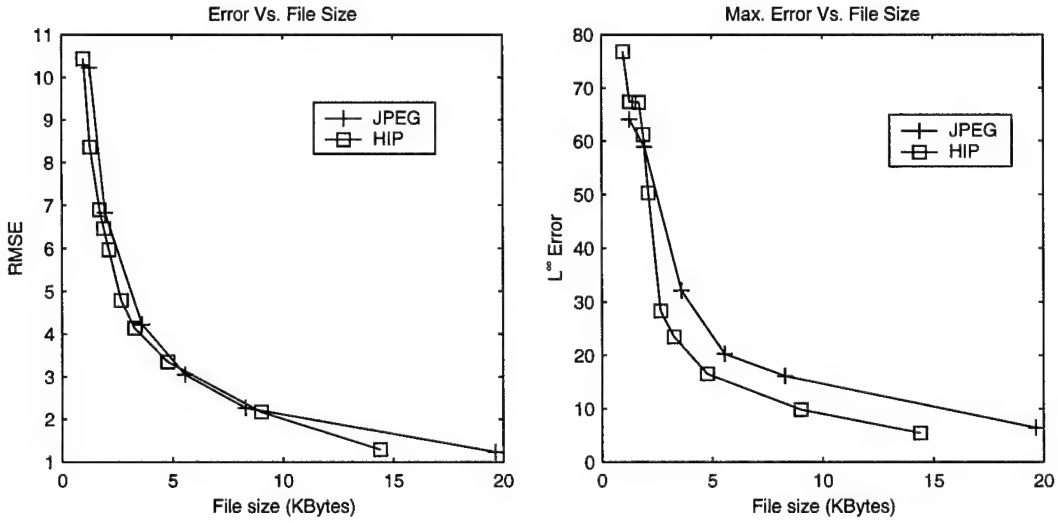

Figure 8. Error vs. file size for HIP compression algorithm and JPEG. The left plot shows root-mean-squared error, while the right plot is maximum error.

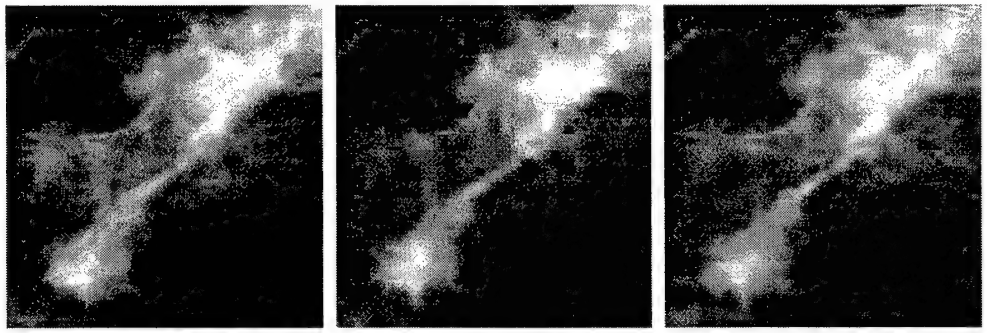

Figure 9. Detail of compressed images. Left is a piece of the original image, center is the corresponding piece of the JPEG-compressed image, and right is the corresponding piece of the HIP compressed image. The JPEG and HIP compression parameters were chosen to give obvious distortion and nearly equal file sizes.

Figure 8 shows the root-mean-squared and maximum errors versus the size of the resulting compressed file, respectively. This is for one randomly-chosen mass ROI image, which was not part of the training set of the HIP model. The HIP algorithm gives mean errors that are comparable to JPEG, and these limited results suggest that its maximum errors are a little lower than with JPEG. It is perhaps not surprising, since the HIP model was fit to similar data while JPEG is intended to be general, but it demonstrates the potential. Compressed and uncompressed images are shown in Figure 9.

f. Microcalcification Classification

We also spent some time applying HIP models to the problem of classifying ROIs as microcalcification clusters or false positives. This differs somewhat from our previous approaches, in which we attempted to detect the individual calcifications, then classify the ROI based on the number of calcifications. The data was provided by Prof. Robert Nishikawa at the University of Chicago. We trained the positive model on 42 ROIs chosen at random from the positives, and the negative model on 88 negatives chosen at random from the full set of negatives. This left 42 positives and 87 negatives for testing. Again, while searching for the best HIP architecture we ran out of computer memory and training became very slow, due to the complexity of the model, before the AIC cost reached a minimum. The resulting ROC curve on the test data had A_z=0.68.

g. Microcalcification synthesis

Applying the trained models to synthesizing microcalcification cluster ROIs gave results like those shown in Figures 10 and 11. Figure 10 shows images generated by HIP models chosen by splitting mixture components first, while the images in Figure 11 were generated by models chosen by splitting the hierarchy components first. Note that all of these images were generated with the same underlying stream of random numbers.

The isolated blobs in Figure 10, especially in the negative image, seem to be due to the poor ability to represent spatial patterns. Otherwise, in both cases the negatives seem to be somewhat smoother, but that was not always the case for other images generated by the models.

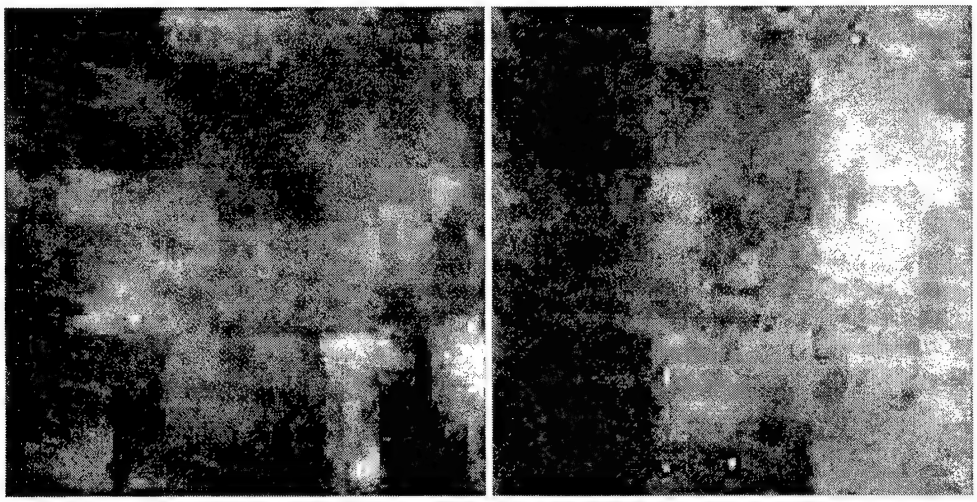

Figure 10. Synthetic microcalcification ROIs generated by HIP models. The left image is from the positive model, while the right is from the negative model. These were generated by HIP models with many mixture components but little information passed between levels by the hidden labels.

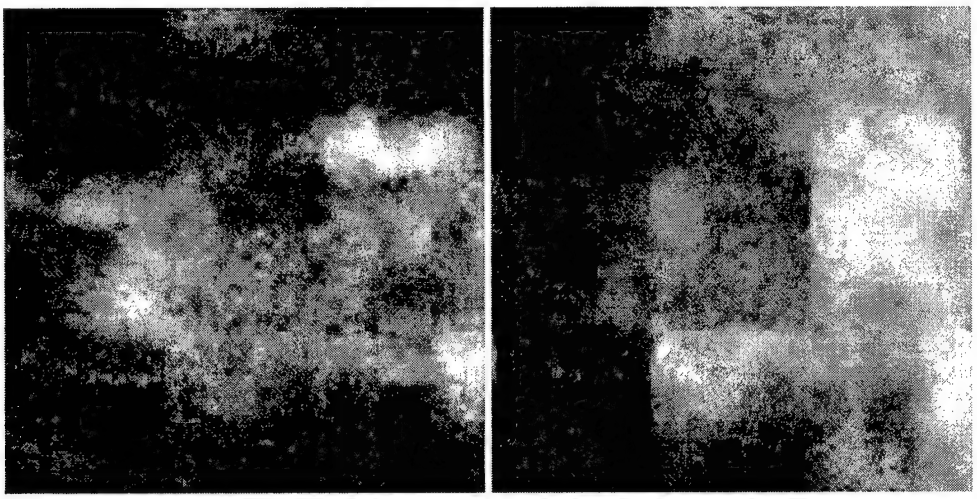

Figure 11. Synthetic microcalcification ROIs generated by HIP models. The left image is from the positive model, while the right is from the negative model. By contrast with the previous figure, these were generated by HIP models with few mixture components but many hierarchy labels, i.e., a great deal of information is passed between levels by the hidden labels.

3. HIP Conclusions

We have developed a class of image probability models we call hierarchical image probability or HIP models. We showed that image distributions can be exactly represented as products over pyramid levels of distributions of sub-sampled feature images conditioned on coarser-scale image information. We argued that hidden variables are needed to capture long-range dependencies while allowing us to further factor the distributions over position. In our current model some of the hidden variables act as indices of mixture components while others condition these mixture component indices and carry information from one level to the next higher resolution level. The resulting model is very similar to the Hidden Markov Tree models, but allows modeling somewhat more general image structures. Because they are models of probability distributions over images, these kinds of models can be used for a wide range of image processing tasks besides classification, e.g., compression, noise-suppression, up-sampling, error correction, etc. Here we presented results for mammographic image analysis, including classification, synthesis, and compression.

However there are obviously other modalities and medical application areas were HIP models would be useful. One in particular is multi-modal fusion, where the problem is to bring a set of images, acquired using different imaging modalities, into alignment. One method that has demonstrated particularly good performance uses mutual information as an objective criterion [22]. The computation of mutual information requires an estimate of entropies, which in turn requires an estimate of the underlying densities of the images. The HIP model potentially provides a framework for learning those densities.

For classification of masses and microcalcifications the HIP models have not been as good as our HPNN neural network classifiers. The cause seems to be that the HIP model is learning a much more complex task, density estimation on image space, as opposed to the simpler task of estimating class probability from the image. Thus the HIP models are much more complex. The information theoretic criteria for choosing between models of different complexity seem to be useful, but we have repeatedly found that more complex models would give better information theoretic cost, when we could not train those more complex models due to limited computer memory and speed. Thus there is room for improvement here in the future. Another path toward better performance is to modify the HIP models. The tree structure of the hidden variables is far from optimal, correlating some neighbors while leaving others much more independent. The only reason for using trees over positions and pyramid levels is computational tractability. Modifying the tree structure will require approximate methods, but may well be worth the difficulties.

III. Key Research Accomplishments

1. Application of hierarchical pyramid neural network (HPNN) to mammographic mass detection. Results show a 51% reduction in false positive rate of The UofC CAD system for mass detection without loss in sensitivity.
2. Development of the hierarchical image probability (HIP) model for mammographic CAD. HIP is a generative model that allows for computing confidence measures based on the training data–an element that is often absent from CAD systems. More importantly, its structure is well-suited for application of MDL model selection techniques.
3. We have developed search strategies and algorithms for selecting a HIP architecture using MDL and AIC information theoretic criteria.
4. HIP models selected by information theoretic criteria for mass detection reduced the false positive rate of the UofC CAD system for mass detection by 30% without loss in sensitivity. We also applied HIP models to the detection of microcalcification clusters.
5. We showed that selecting wavelet packet bases using an information theoretic criterion (entropy) gives an image representation that allows a more computationally efficient HIP architecture. In particular the model requires much less memory.
6. We have shown that different information theoretic measures track the HIP generalization performance and thus offer good criteria for model selection.
7. We have demonstrated the utility of the HIP architecture for identifying novel ROIs. This novelty detection is useful for defining confidence measures for the classifier.
8. We have demonstrated the utility of the HIP architecture for synthesizing new positive and negative mammographic ROIs. We have discussed how synthesis can be used to gain an intuitive understanding of the structure that is captured by the model.
9. We have demonstrated the utility of the HIP architecture for image compression.

IV. Reportable Outcomes

1. Disclosure/Patent Application "Hierarchical Image Probability Models", March 1999.

2. Clay Spence, Lucas Parra and Paul Sajda, "Mammographic mass detection with a hierarchical image probability (HIP) model", in *Medical Imaging 2000: Image Processing*, Kenneth M. Hanson, Editor, Proceedings of SPIE Vol. 3979, pp. 990–997 (2000).
3. Presentation "Hierarchical Pattern Recognition for Mammographic CAD", University of Pennsylvania, November 1998.
4. Invited talk at Columbia University Medical School "Hierarchical Neural Networks for Object Recognition: Applications to Mammographic Computer-aided Diagnosis", June 2000
5. DoD Era of Hope meeting (poster), "A Hierarchical Image Probability Model for Mammographic Mass Detection", June 2000
6. Invited lecture at The University Of Pennsylvania, Department of Bioengineering "Computer Assisted Diagnosis for Mammography", November 1999
7. NIMA/DARPA Medical Dual-use project ($1.8M). Focus on developing dual-use technology for medical and military applications. Medical areas include breast cancer, lung cancer, retinal disease and neurological disease.
8. Clay Spence, Lucas Parra and Paul Sajda, "Hierarchical Image Probability (HIP) Models." In the Proceedings of *ICIP 2000*, the IEEE International Conference on Image Processing.
9. Clay Spence, Lucas Parra and Paul Sajda, "Detection, synthesis and compression in mammographic image analysis using a Hierarchical Image Probability (HIP) model." Submitted to MMBIA 2001, the *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis.*
10. Paul Sajda and Clay Spence, "Learning Contextual Relationships in Mammograms using a Hierarchical Pyramid Neural Network." Submitted to *IEEE Trans. Medical Imaging*. (Accepted subject to minor revisions.)

V. Conclusion

We have demonstrated the utility of hierarchical pattern recognizers for improving the performance of CAD systems for mass detection. Mass detection is currently the more difficult problem in mammographic CAD (compared to microcalcification detection). For CAD systems to gain clinical acceptance, false positives must be significantly reduced without loss in sensitivity. On a small research database, application of our HPNN model has resulted in a 51% reduction in false positive rate of the UofC CAD system for mass detection. However the HPNN models that we have trained are not well-suited to objective model selection techniques, such as MDL. Since objective model selection is often critical to maximizing the performance of a pattern recognizer, we developed a new hierarchical pattern recognition framework that we call the hierarchical image probability (HIP) model.

We have developed search strategies for applying information theoretic criteria to the problem of selecting the best label architecture for a HIP model. Furthermore, we demonstrated that these criteria correlate well with generalization performance of the classifiers.

Our results show that HIP models selected using these criteria can reduce false positive rates by 30% for a data set constructed using The University of Chicago CAD mass detection system. It appears that further improvements are likely simply by using more complex HIP models.

We have used the generative structure of the HIP model to detect novel examples—examples that significantly differ from the training data. Novelty detection can be used to generate confidence measures and we have shown how these confidence measures can be used to improve ROC performance. In practice such examples would be rejected as not reliably classifiable by the models. This capability is not shared with classifiers that directly estimate the probability of the class given the image.

We have demonstrated other useful aspects of the generative feature of HIP models. We have sampled positive and negative HIP models for synthesizing ROIs, enabling us to gain an intuition into the structure the HIP model learns for representing the two classes. We also developed a simple algorithm for compressing images using HIP models, and showed that it performs a little better than JPEG on mammographic regions of interest.

A. "So What" Section

Statistical pattern recognition is a key element in any mammographic computer-aided diagnosis system. Hierarchical pattern recognizers are particularly useful since they are capable of exploiting contextual and multi-resolution information for detecting clinically significant objects. Most statistical pattern recognizers that have been previously developed for mammographic CAD have been trained to estimate the probability of the class, e.g., mass or non-mass, given the image or some features extracted from the image. By contrast HIP models are trained to estimate the probability distribution of images. This gives HIP models many attractive features. One could use HIP for detection/classification in the usual way by training a distribution for each object class and using Bayes' rule to get the class probability. We have reported our initial results for this application of HIP in this report.

Even though our original motivation for this model was to develop a framework for hierarchical pattern recognition which could exploit techniques in MDL model selection, there are other attractive features of the HIP framework which could have a major impact on the design and development of mammographic CAD systems. Since HIP computes the probability density at the input image, we could attempt to detect unusual images and reject them rather than trust the classifier; something that is not possible with models of the class probability. Building confidence measures into CAD systems is an open area of research and the HIP model provides a mechanism by which to generate these measures. In fact we have shown results illustrating how novelty detection can be used to improve the ROC performance of CAD systems.

The HIP model has applications other than detection/classification, and can be used in these applications without further training. Since the HIP model is a generative model, one can use it to compress data, given the probability distribution of the objects of interest. If one wants lossless compression of a digital mammogram one need only train a HIP model for a set of mammographic images and then use the probability model to compress the data. More interesting is the application of HIP for lossy compression. In that case, one might train a HIP model on clinically significant objects, such as mammographic masses, since those are the parts of the image one would like to preserve–i.e. have minimal distortion and compression artifacts. The entire image can then be compressed using this model. Though there will be loss over regions of the mammogram which do not fit the model, those regions of clinical significance will be preserved since they will have a good fit to the probability model and require very few bits for compression.

In all of these applications, an essential role was played by the information theoretic algorithms for architecture selection. These have proven themselves as reliable and computationally efficient guides to the generalization capability of the different classifier architectures.

VI. References

[1] J. A. Rissanen. A universal prior for integers and estimation of minimum description length. *Annals of Statistics*, 11(2):416–431, 1983.

[2] R. M. Nishikawa, R. C. Haldemann, J. Papaioannou, M. L. Giger, P. Lu, R.A. Schmidt, D. E. Wolverton, U. Bick, and K. Doi. Initial experience with a prototype clinical intelligent mammography workstation for computer-aided diagnosis. In *Medical Imaging 1995*, Murray H. Loew and Kenneth M. Hanson, editors, Proceedings of SPIE Vol. 2434, 65-71, 1995.

[3] P. Sajda, C. Spence and L. Parra, Applications of information theory to improve computer-aided diagnosis Systems, Year 1 Report, *DoD Breast Cancer Program*, DAMD17-98-1-8061, July 1999

[4] J. A. Rissanen. Information theory and neural nets. In Smolensky, Mozer, and Rumelhart, editors, *Mathematical Perspectives on Neural Networks*, pages 567–602, 1996.

[5] Paul Sajda, Clay D. Spence, John C. Pearson, and Robert M. Nishikawa. Exploiting context in mammograms: A hierarchical neural network for detecting microcalcifications. In Murray H. Loew and Kenneth M. Hanson, editors, *Medical Imaging 1996 — Image Processing*, volume 2710, pages 733–742, P.O. Box 10, Bellingham WA 98227-0010, 1996. SPIE.

[6] Clay D. Spence, Paul Sajda, and Robert M. Nishikawa. Dealing with uncertainty and error in truth data when training neural networks for computer-aided diagnosis applications. In Heinz U. Lemke, Michael W. Vannier, and Kiyonari Inamura, editors, *CAR '97: Proceedings of the 11th International Symposium on Computer Assisted Radiology and Surgery*, pages 352–357, Amsterdam, 1997. Elsevier.

[7] Y. Le Cun, B. Boser, J. S. Denker, and D. Henderson. Handwritten digit recognition with a back-propagation network. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems* 2, pages 396–404, 2929 Campus Drive, San Mateo, CA 94403, 1991. Morgan-Kaufmann Publishers.

[8] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.

[9] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, PAMI-6(6):194–207, November 1984.

[10] Rama Chellappa and S. Chatterjee. Classification of textures using Gaussian Markov random fields. *IEEE Trans. ASSP*, 33:959–963, 1985.

[11] Jeremy S. De Bonet and Paul Viola. Texture recognition using a non-parametric multi-scale statistical model. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 1998.

[12] J. S. De Bonet, P. Viola, and J. W. Fisher III. Flexible histograms: A multiresolution target discrimination model. In E. G. Zelnio, editor, *Proceedings of SPIE*, volume 3370, 1998.

[13] Mark R. Luettgen and Alan S. Willsky. Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination. *IEEE Trans. Image Proc.*, 4(2):194–207, 1995.

[14] Robert W. Buccigrossi and Eero P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. Technical Report 414, U. Penn. GRASP Laboratory, 1998. Available at ftp://ftp.cis.upenn.edu/pub/eero/buccigrossi97.ps.gz.

[15] Matthew S. Crouse, Robert D. Nowak and Richard G. Baraniuk, Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. Sig. Proc.* 46(4), pp. 886–902. 1998.

[16] Hui Cheng and Charles A. Bouman, Multiscale Bayesian segmentation using a trainable context model. *IEEE Trans. Image Proc.* 10(4), pp. 511–525. 2001.

[17] Charles A. Bouman and Michael Schapiro, A multiscale random field model for Bayesian image segmentation. *IEEE Trans. Image Proc.* 3(2), pp. 162–177. 1994.

[18] Edward Adelson, Eero Simoncelli and Rajesh Hingorani, "Orthogonal pyramid transforms for image coding." In *Visual Communication and Image Processing II*, SPIE, Bellingham WA, 1987.

[19] Martin J. Wainwright and Eero Simoncelli, "Scale mixtures of Gaussians and the statistics of natural images." In *Advances in Neural Information Processing Systems 12*. Sara A. Solla, Todd K. Lean, and Klaus-Robert Müller, eds. MIT Press, 2000.

[20] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E. Hinton. SMEM algorithm for mixture models. In Michael S. Kearns, Sara A. Solla, and David A. Cohn, editors, *Advances in Neural Information Processing Systems 1*1, pages 599–605, Massachusetts Institute of Technology, Cambridge, MA 02142, 1999. MIT Press.

[21] Eero Simoncelli, William T. Freeman and Edward Adelson, "Shiftable multiscale transforms." *IEEE Trans. Info. Theory* 38(2), pp 587–607. March 1992.

[22] C. M. Bishop, Novelty detection and neural network validation. *IEE Proceedings: Vision, Image and Signal Processing*, 141 (4), 217-222, 1994.

[23] Naoki Saito, "Local feature extraction and its applications using a library of bases." Ph.D. dissertation, Yale University, December 1994.

[24] Paul Viola and William M. Wells III, "Alignment by maximization of mutual information." *Intern. J. Computer Vision*, 24(2), pp 137–154 (1997).

VII. Appendices

Attached papers:

C. Spence, L. Parra, and P. Sajda, "Mammographic mass detection with a hierarchical image probability (HIP) model," in *Medical Imaging 2000: Image Processing*, Kenneth M. Hanson, Editor, Proceedings of SPIE Vol. 3979, 990-997 (2000).

Clay Spence, Lucas Parra and Paul Sajda, "Hierarchical Image Probability (HIP) Models." In the Proceedings of *ICIP 2000*, the IEEE International Conference on Image Processing.

Clay Spence, Lucas Parra and Paul Sajda, "Detection, synthesis and compression in mammographic image analysis using a Hierarchical Image Probability (HIP) model." Submitted to MMBIA 2001, the *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis.*

Paul Sajda and Clay Spence, "Learning Contextual Relationships in Mammograms using a Hierarchical Pyramid Neural Network." Submitted to *IEEE Trans. Medical Imaging*. (Accepted subject to minor revisions.)

Mammographic mass detection with a hierarchical image probability (HIP) model

Clay Spence, Lucas Parra, and Paul Sajda

Sarnoff Corporation CN5300 Princeton, NJ 08543-5300

ABSTRACT

We formulate a model for probability distributions on image spaces. We show that any distribution of images can be factored exactly into conditional distributions of feature vectors at one resolution (pyramid level) conditioned on the image information at lower resolutions. We would like to factor this over positions in the pyramid levels to make it tractable, but such factoring may miss long-range dependencies. To fix this, we introduce hidden class labels at each pixel in the pyramid. The result is a hierarchical mixture of conditional probabilities, similar to a hidden Markov model on a tree. The model parameters can be found with maximum likelihood estimation using the EM algorithm. We have obtained encouraging preliminary results on the problems of detecting masses in mammograms.

Keywords: Mammography, CAD, Image Probability

1. INTRODUCTION

Many approaches to object recognition in images estimate Pr(class | image). By contrast, a model of the probability distribution of images, Pr(image), has many attractive features. We could use this for object recognition in the usual way by training a distribution for each object class and using Bayes' rule to get Pr(class | image) = Pr(image | class) Pr(class)/ Pr(image). Clearly there are many other benefits of having a model of the distribution of images, since any kind of data analysis task can be approached using knowledge of the distribution of the data. For classification we could attempt to detect unusual examples and reject them, rather than trusting the classifier's output. We could also compress, interpolate, suppress noise, extend resolution, fuse multiple images, etc.

Many image analysis algorithms use probability concepts, but few treat the distribution of images. One of the few examples of image distribution models was constructed by Zhu, Wu and Mumford.[1] They compute the maximum entropy distribution given a set of statistics for some features, which seems to work well for textures but it is not clear how well it will model the appearance of more structured objects.

There are several algorithms for modeling the distributions of features extracted from the image, instead of the image itself. The Markov Random Field (*MRF*) models are an example of this line of development; see, e.g., References 2,3. However, they tend to be very computationally expensive.

In De Bonet and Viola's flexible histogram approach,[4,5] features are extracted at multiple image scales, and the resulting feature vectors are treated as a set of independent samples drawn from a distribution. The distribution of feature vectors is then modeled using Parzen windows. This has given good results, but the feature vectors from neighboring pixels are treated as independent when in fact they share exactly the same components from lower-resolutions. To fix this one might build a model in which the features at one pixel of one pyramid level condition the features at each of several child pixels at the next higher-resolution pyramid level. The multiscale stochastic process (*MSP*) methods do exactly that. Luettgen and Willsky,[6] for example, applied a scale-space auto-regression (AR) model to texture discrimination. They use a quadtree or quadtree-like organization of the pixels in an image pyramid, and model the features in the pyramid as a stochastic process from coarse-to-fine levels along the tree. The variables in the process are hidden, and the observations are sums of these hidden variables plus noise. The Gaussian distributions are a limitation of MSP models. The result is also a model of the probability of the observations on the tree, not of the image.

All of these methods seem well-suited for modeling texture, but it is unclear how one might build models to capture the appearance of more structured objects. We will argue below that the presence of objects in images can make local conditioning like that of the flexible histogram and MSP approaches inappropriate. In the following we

E-mail: {cspence, lparra, psajda}@sarnoff.com

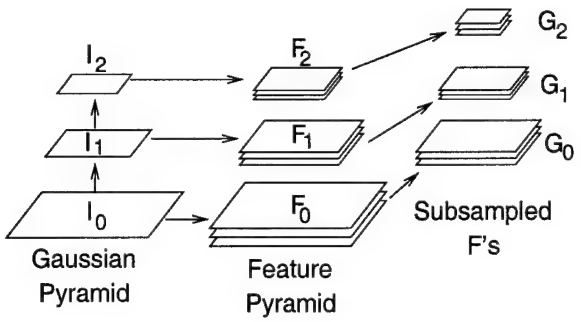

Figure 1. Pyramids and feature notation.

present a model for probability distributions of images, in which we try to move beyond texture modeling. This hierarchical image probability (*HIP*) model is similar to a hidden Markov model on a tree, and can be learned with the EM algorithm. In preliminary tests of the model on classification tasks the performance was comparable to that of other algorithms.

2. COARSE-TO-FINE FACTORING OF IMAGE DISTRIBUTIONS

Our goal will be to write the image distribution in a form similar to $\Pr(I) \sim \Pr(\mathbf{F}_0 \,|\, \mathbf{F}_1) \Pr(\mathbf{F}_1 \,|\, \mathbf{F}_2) \ldots$, where $\mathbf{F}_l$ is the set of feature images at pyramid level l. We expect that the short-range dependencies can be captured by the model's distribution of individual feature vectors, while the long-range dependencies can be captured somehow at low resolution. The large-scale structures affect finer scales by the conditioning.

In fact we can prove that a coarse-to-fine factoring like this is correct. From an image I we build a Gaussian pyramid (repeatedly blur-and-subsample, with a Gaussian filter). Call the l-th level I_l, e.g., the original image is I_0 (Figure 1). From each Gaussian level I_l we extract some set of feature images $\mathbf{F}_l$. Sub-sample these to get feature images $\mathbf{G}_l$. Note that the images in $\mathbf{G}_l$ have the same dimensions as I_{l+1}. We denote by $\tilde{\mathbf{G}}_l$ the set of images containing I_{l+1} and the images in $\mathbf{G}_l$. We further denote the mapping from I_l to $\tilde{\mathbf{G}}_l$ by $\tilde{\mathcal{G}}_l$.

Suppose now that $\tilde{\mathcal{G}}_0 : I_0 \mapsto \tilde{\mathbf{G}}_0$ is invertible. Then we can think of $\tilde{\mathcal{G}}_0$ as a change of variables. If we have a distribution on a space, its expressions in two different coordinate systems are related by multiplying by the Jacobian. In this case we get $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\tilde{\mathbf{G}}_0)$. Since $\tilde{\mathbf{G}}_0 = (\mathbf{G}_0, I_1)$, we can factor $\Pr(\tilde{\mathbf{G}}_0)$ to get $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\mathbf{G}_0 \,|\, I_1) \Pr(I_1)$. If $\tilde{\mathcal{G}}_l$ is invertible for all $l \in \{0, \ldots, L-1\}$ then we can simply repeat this change of variable and factoring procedure to get

$$\Pr(I) = \left[\prod_{l=0}^{L-1} |\tilde{\mathcal{G}}_l| \Pr(\mathbf{G}_l \,|\, I_{l+1}) \right] \Pr(I_L) \tag{1}$$

This is a very general result, valid for all $\Pr(I)$, no doubt with some rather mild restrictions to make the change of variables valid. The restriction that $\tilde{\mathcal{G}}_l$ be invertible is strong, but many such feature sets are known to exist, e.g., most wavelet transforms on images.

3. THE NEED FOR HIDDEN VARIABLES

For the sake of tractability we want to factor $\Pr(\mathbf{G}_l \,|\, I_{l+1})$ over positions, something like

$$\Pr(I) \sim \prod_l \prod_{x \in I_{l+1}} \Pr\big(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x)\big)$$

where $\mathbf{g}_l(x)$ and $\mathbf{f}_{l+1}(x)$ are the feature vectors at position x. The dependence of $\mathbf{g}_l$ on $\mathbf{f}_{l+1}$ expresses the persistence of image structures across scale, e.g., an edge is usually detectable as such in several neighboring pyramid levels. The flexible histogram and MSP methods share this structure.

While it may be plausible that $\mathbf{f}_{l+1}(x)$ has a strong influence on $\mathbf{g}_l(x)$, a model distribution with this factorization and conditioning cannot capture some properties of real images. Objects in the world cause correlations and non-local dependencies in images. For example, the presence of a particular object might cause a certain kind of texture to be visible at level l. Usually local features $\mathbf{f}_{l+1}$ by themselves will not contain enough information to infer the object's presence, but the entire image I_{l+1} at that layer might. Thus $\mathbf{g}_l(x)$ is influenced by more of I_{l+1} than the local feature vector.

Similarly, objects create long-range dependencies. For example, an object class might result in a kind of texture across a large area of the image. If an object of this class is always present, the distribution may factor, but if such objects aren't always present and can't be inferred from lower-resolution information, the presence of the texture at one location affects the probability of its presence elsewhere.

We introduce hidden variables to represent the non-local information that is not captured by local features. They should also constrain the variability of features at the next finer scale. Denoting them collectively by A, we assume that conditioning on A allows the distributions over feature vectors to factor. In general, the distribution over images becomes

$$\Pr(I) \propto \sum_A \left\{ \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Pr\big(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x), A\big) \Pr(A \,|\, I_{L+1}) \right\} \Pr(I_{L+1}). \tag{2}$$

As written this is absolutely general, so we need to be more specific. In particular we would like to preserve the conditioning of higher-resolution information on coarser-resolution information, and the ability to factor over positions.

As a first model we have chosen the following structure for our HIP model:*

$$\Pr(I) \propto \sum_{A_0,\ldots,A_L} \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Big[\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \Pr(a_l \,|\, a_{l+1}, x) \Big] \tag{3}$$

To each position x at each level l we attach a hidden discrete index or label $a_l(x)$. The resulting label image A_l for level l has the same dimensions as the images in $\tilde{\mathbf{G}}_l$.

Since $a_l(x)$ codes non-local information we can think of the labels A_l as a segmentation or classification at the l-th pyramid level. By conditioning $a_l(x)$ on $a_{l+1}(x)$, we mean that $a_l(x)$ is conditioned on a_{l+1} at the *parent* pixel of x. This parent-child relationship follows from the sub-sampling operation. For example, if we sub-sample by two in each direction to get $\mathbf{G}_l$ from $\mathbf{F}_l$, we condition the variable a_l at (x, y) in level l on a_{l+1} at location $(\lfloor x/2 \rfloor, \lfloor y/2 \rfloor)$ in level $l+1$ (Figure 2). This gives the dependency graph of the hidden variables a tree structure. Such a probabilistic tree of discrete variables is sometimes referred to as a belief network. By conditioning child labels on their parents information propagates though the layers to other areas of the image while accumulating information along the way.

For the sake of simplicity we've chosen $\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l)$ to be normal with mean $\bar{\mathbf{g}}_{l,a_l} + M_{a_l}\mathbf{f}_{l+1}$ and covariance Σ_{a_l}, that is,

$$\Pr(\mathbf{g} \,|\, \mathbf{f}, a) = \mathcal{N}(\mathbf{g}, M_a\mathbf{f} + \bar{\mathbf{g}}_a, \Lambda_a) \tag{4}$$

4. EM ALGORITHM

Due to the tree structure, the belief network for the hidden variables is relatively easy to train with an EM algorithm. The expectation step (summing over a_l's) can be performed directly. If we had chosen a more densely-connected structure with each child having several parents, we would need either an approximate algorithm or Monte Carlo techniques. The expectation is weighted by the probability of a label or a parent-child pair of labels given the image. This can be computed in a fine-to-coarse-to-fine procedure, i.e. working from leaves to the root and then back out to the leaves. The method is based on belief propagation.[7]

*In principle there is also a factor of $\Pr(I_{L+1})$. In many cases I_{L+1} will be a single pixel that is approximately the mean brightness in the image. We ignore this, which is equivalent to assuming that $\Pr(I_{L+1})$ is flat over some range. In this case $\mathbf{f}_{L+1}$ is zero for typical features. In addition, there is no hidden variable a_{L+1}. If we combine these considerations we see that the $l = L$ factor should be read as $\prod_x \Pr(\mathbf{g}_L \,|\, a_L, x) \Pr(a_L, x)$.

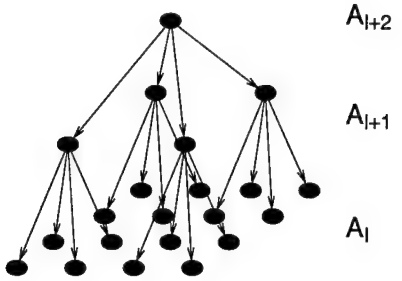

Figure 2. Tree structure of the conditional dependency between hidden variables in the HIP model. With subsampling by two, this is sometimes called a quadtree structure.

Once we can compute the expectations, the normal distribution makes the M-step tractable; we simply compute the updated $\bar{\mathbf{g}}_{a_l}$, Σ_{a_l}, M_{a_l}, and $\Pr(a_l \,|\, a_{l+1})$ as combinations of various expectation values.

In order to apply the EM algorithm, we need to choose a parameterization for the model. The parameterization of $\Pr(\mathbf{g} \,|\, \mathbf{f}, a)$ is given above in Equation 4. For $\Pr(a_l \,|\, a_{l+1})$ we use the parameterization

$$\Pr(a_l \,|\, a_{l+1}) = \frac{\pi_{a_l, a_{l+1}}}{\sum_{a_l} \pi_{a_l, a_{l+1}}} \tag{5}$$

in order to ensure proper normalization.

Below, we denote the new parameter values computed during the t-th maximization step as θ^{t+1} and the old values as θ^t.

4.1. MAXIMIZATION

Maximizing the expectation of the likelihood over the hidden variables with respect to the model parameters gives the following update formulae:

$$\pi^{t+1}_{a_l, a_{l+1}} = \sum_x \Pr(a_l, a_{l+1}, x | I, \theta^t), \tag{6}$$

$$M^{t+1}_{a_l} = \left(\langle \mathbf{g}_l \mathbf{f}^T_{l+1} \rangle_{t,a_l} - \langle \mathbf{g}_l \rangle_{t,a_l} \langle \mathbf{f}^T_{l+1} \rangle_{t,a_l} \right) \left(\langle \mathbf{f}_{l+1} \mathbf{f}^T_{l+1} \rangle_{t,a_l} - \langle \mathbf{f}_{l+1} \rangle_{t,a_l} \langle \mathbf{f}^T_{l+1} \rangle_{t,a_l} \right)^{-1}, \tag{7}$$

$$\bar{\mathbf{g}}^{t+1}_{a_l} = \langle \mathbf{g}_l \rangle_{t,a_l} - M^{t+1}_{a_l} \langle \mathbf{f}_{l+1} \rangle_{t,a_l}, \tag{8}$$

and

$$\Lambda^{t+1}_{a_l} = \left\langle \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1} \right) \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1} \right)^T \right\rangle_{t,a_l} - \bar{\mathbf{g}}^{t+1}_{a_l} \bar{\mathbf{g}}^{t+1\,T}_{a_l}. \tag{9}$$

Here the brackets $\langle . \rangle_{t,a_l}$ denotes the expectation value

$$\langle X \rangle_{t,a_l} = \frac{\sum_x \Pr(a_l, x \,|\, I, \theta^t) X(x)}{\sum_x \Pr(a_l, x \,|\, I, \theta^t)}. \tag{10}$$

4.2. EXPECTATION

In the E-step we need to compute the probabilities of pairs of labels from neighboring layers $\Pr(a_l, a_{l+1}, x_l \,|\, I, \theta^t)$ for given image data. But note that in all occurrences of the reestimation equations, i.e. (5,6) and (10), we need that quantity only up to an overall factor. We can choose that factor to be $\Pr(I|\theta^t)$ and can therefore compute $\Pr(a_l, a_{l+1}, x_l, I|\theta^t)$ instead using

$$\Pr(a_l, a_{l+1}, x \,|\, I, \theta^t) \Pr(I \,|\, \theta^t) = \Pr(a_l, a_{l+1}, x, I \,|\, \theta^t) = \sum_{A \setminus a_l(x), a_{l+1}(x)} \Pr(I, A|\theta^t) \tag{11}$$

The computation of these quantities can be cast as recursion formulae, defined in terms of quantities u and d, which approximately represent upwards and downwards propagating probabilities. The recursion formulae are

$$u_l(a_l, x) = \Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \prod_{x' \in \mathsf{Ch}(x)} \tilde{u}_{l-1}(a_l, x') \tag{12}$$

$$\tilde{u}_l(a_{l+1}, x) = \sum_{a_l} \Pr(a_l | \, a_{l+1}) u_l(a_l, x) \tag{13}$$

$$d_l(a_l, x) = \sum_{a_{l+1}} \Pr(a_l | \, a_{l+1}) \tilde{d}_l(a_{l+1}, x) \tag{14}$$

$$\tilde{d}_l(a_{l+1}, x) = \frac{u_{l+1}(a_{l+1}, \mathsf{Par}(x))}{\tilde{u}_l(a_{l+1}, x)} d_{l+1}(a_{l+1}, \mathsf{Par}(x)) \tag{15}$$

The upward recursion relations (12–13) are initialized at $l = 0$ with $u_0(a_0, x) = \Pr(\mathbf{g} \,|\, \mathbf{f}_1, a_0, x)$ and end at $l = L$. At layer L Equation 13 reduces to $\tilde{u}_L(a_{L+1}, x) = \tilde{u}_L(x)$.† Since we do not model any further dependencies beyond layer L, the pixels at layer L are assumed independent. Considering the definition of u, it is evident that the product of all $\tilde{u}_L(x)$ coincides with the total image probability,

$$\Pr(I|\theta^t) = \prod_{x \in I_L} \tilde{u}_L(x) = u_{L+1}. \tag{16}$$

The downward recursion (14 - 15) can be executed, starting with equation (15) at $l = L$ with $d_{L+1}(a_{L+1}, x) = d_{L+1}(x) = 1$.† The downwards recursion ends at $l = 0$ with equation (14).

We can now compute (11) as

$$\Pr(a_l, a_{l+1}, x, I \,|\, \theta^t) = u_l(a_l, x) \tilde{d}_l(a_{l+1}, x) \Pr(a_l | a_{l+1}) \tag{17}$$

$$\Pr(a_l, x, I \,|\, \theta^t) = u_l(a_l, x) d_l(a_l, x) \tag{18}$$

Obviously computations (12–18) in the E-step at iteration t need to be completed with fixed parameters θ^t.

Because of the dependence of $\mathbf{g}_l$ on $\mathbf{f}_{l+1}$, these u's and d's are not, in general, actual probabilities. In spite of this it can be shown that these recursion relations are correct.

5. EXPERIMENTS

5.1. CLASSIFICATION OF VEHICLES IN SAR IMAGERY

Though not a medical imaging problem, we first present the results of our experiments on synthetic aperture radar (*SAR*) imagery, since SAR imagery is noisy and involves detecting an extended textured object, much like a breast mass and many other medical imaging problems. The problem was to discriminate between three target classes in the MSTAR public targets data set, to compare with the results of the flexible histogram approach of De Bonet, et al.[5] We trained three HIP models, one for each of the target vehicles BMP-2, BTR-70 and T-72 (Figure 3). As in Reference 5 we trained each model on ten images of its class, one image for each of ten aspect angles, spaced approximately 36° apart. We trained one model for all ten images of a target, whereas De Bonet et al trained one model per image.

We first tried discriminating between vehicles of one class and other objects by thresholding $\log \Pr(I \,|\, \text{class})$, i.e., no model of other objects is used. In essence this discriminates simply by judging whether an image looks sufficiently similar to the training examples. For the tests, the other objects were taken from the test data for the two other vehicle classes, plus seven other vehicle classes. There were 1,838 image from these seven other classes, 391 BMP2 test images, 196 BTR70 test images, and 386 T72 test images. The resulting ROC curves are shown in Figure 4a.

We then tried discriminating between pairs of target classes using HIP model likelihood ratios, i.e., $\log \Pr(I \,|\, \text{class1}) - \log \Pr(I \,|\, \text{class2})$. Here we could not use the extra seven vehicle classes. The resulting ROC curves are shown in Figure 4b. The performance is comparable to that of the flexible histogram approach.

†The (non-existent) label a_{L+1} can be thought of as a label with a single possible value, which is always set. The conditional $\Pr(a_L | \, a_{L+1})$ turns then into a prior $\Pr(a_L)$

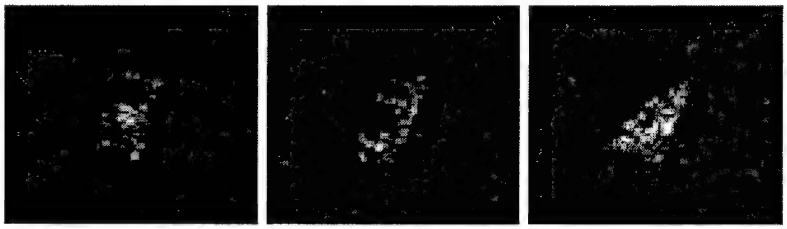

Figure 3. SAR images of three types of vehicles to be detected.

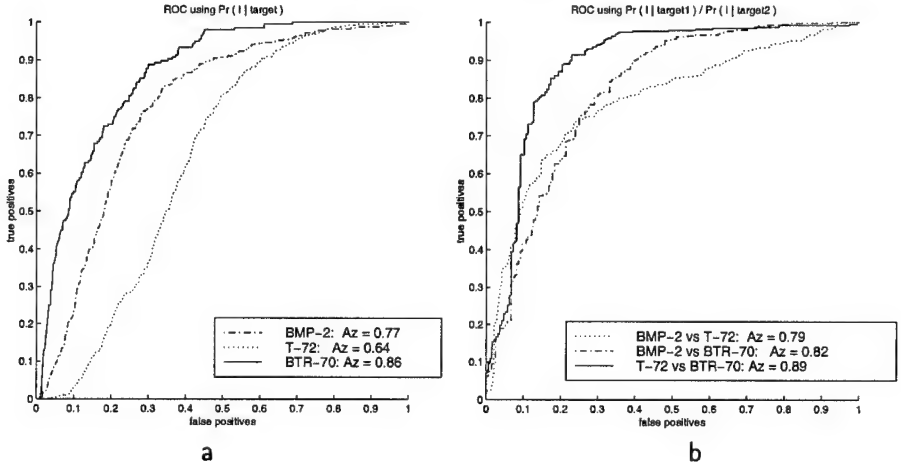

Figure 4. ROC curves for vehicle detection in SAR imagery. (a) ROC curves by thresholding HIP likelihood of desired class. (b) ROC curves for inter-class discrimination using ratios of likelihoods as given by HIP models.

5.2. MASS DETECTION

We applied HIP to the problem of detecting masses in ROIs taken from mammograms, as detected by a CAD system at the University of Chicago. We trained a HIP model of the distribution of positive images on 36 randomly-chosen ROIs that contained masses, and a second HIP model on 48 randomly-chosen ROIs without masses. The likelihood ratio was then used as the test criterion, i.e., a threshold on this ratio is used to decide which ROIs will be called masses. The true and false positive rates as a function of the threshold were measured on a test set with 36 mass and 49 non-mass ROIs.

A search was performed over the number of hidden labels values at each level. The search criterion was the negative log-likelihood on the training data plus the minimum-description-length penalty term, $d\log(N)/2$, where d is the number of model parameters and N is the the number of training examples. The maximum number of labels in a level was bounded (somewhat arbitrarily) at 17, since doubling the number of components in a level at this point was observed to decrease the MDL criterion, but very little, and the computation time would approximately double.

The best architecture had 17, 17, 11, 2, and 1 hidden label in levels 0–4, respectively. For this architecture, A_z was 0.73. This detector had a specificity of 33 % at a sensitivity of 95 %. The ROC curve is shown in Figure 5. While this performance is not as good as we might hope, being worse than our own HPNN classifier,[8] for instance, it demonstrates that the model captures relevant information for classification. We hope that further work, particularly in model and feature selection, will improve on these results.

6. CONCLUSION

We have developed a class of image probability models we call hierarchical image probability or HIP models. To justify these, we showed that image distributions can be exactly represented as products over pyramid levels of distributions of sub-sampled feature images conditioned on coarser-scale image information. We argued that hidden variables are needed to capture long-range dependencies while allowing us to further factor the distributions over position. In our current model the hidden variables act as indices of mixture components. The resulting model is

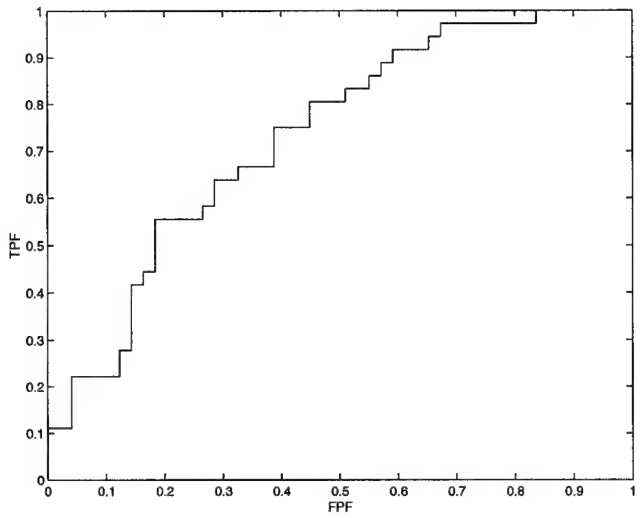

Figure 5. ROC curve for HIP detector of Mass ROIs generated by U. Chicago CAD.

somewhat like a hidden Markov model on a tree. The HIP model can be used for a wide range of image processing tasks besides classification, e.g., compression, noise-suppression, up-sampling, error correction, etc.

There is much room for further work on variations of the specific HIP model presented here. The tree-structured discrete hidden variables lend themselves well to exact marginalization, but they fail to capture certain image properties. For example, contrast level and orientation could be given continuous parameterizations. See, for example, the work of Simoncelli and Wainwright, who developed a very similar model to capture the statistics of contrast level (which they refer to as "scale"), though they did not formulate their model as an image probability.[9] Furthermore, as is well known, the tree structure of the hidden variable dependencies will tend to artificially suppress the statistical dependence between some neighboring pixels, but not others. Allowing multiple parents would alleviate this. Unfortunately, either of these modifications would make it impractical to marginalize over the hidden variables, which is the proper probabilistic procedure. There are approximate alternatives to exact marginalization, which should allow a far wider variety of hidden variable structures.

ACKNOWLEDGEMENTS

We thank Drs. Robert Nishikawa and Maryellen Giger of The University of Chicago for useful discussions and providing the data. This work was supported by the US Department of the Army under grant number DAMD17-98-1-8061. This paper does not necessarily reflect the position or the policy of the US government, and no official endorsement should be inferred.

REFERENCES

1. S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Computation* **9**(8), pp. 1627–1660, 1997.
2. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. PAMI* **PAMI-6**, pp. 194–207, Nov. 1984.
3. R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. ASSP* **33**, pp. 959–963, 1985.
4. J. S. D. Bonet and P. Viola, "Texture recognition using a non-parametric multi-scale statistical model," in *Conference on Computer Vision and Pattern Recognition*, IEEE, 1998.
5. J. S. D. Bonet, P. Viola, and J. W. F. III, "Flexible histograms: A multiresolution target discrimination model," in *Proceedings of SPIE*, E. G. Zelnio, ed., vol. 3370, 1998.
6. M. R. Luettgen and A. S. Willsky, "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination," *IEEE Trans. Image Proc.* **4**(2), pp. 194–207, 1995.

7. M. I. Jordan, ed., *Learning in Graphical Models*, vol. 89 of *NATO Science Series D: Behavioral and Brain Sciences*, Kluwer Academic, 1998.
8. C. D. Spence and P. Sajda, "Applications of multi-resolution neural networks to mammography," in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, eds., pp. 981–988, MIT Press, (Cambridge, MA), 1998.
9. M. J. Wainwright and E. P. Simoncelli, "Scale mixtures of Gaussians and the statistics of natural images," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. Leen, and K.-R. Müller, eds., MIT Press, (Cambridge, MA), 1999.

HIERARCHICAL IMAGE PROBABILITY (HIP) MODELS

Clay Spence, Lucas Parra and Paul Sajda

Sarnoff Corporation
CN5300
Princeton, NJ 08543-5300

ABSTRACT

We formulate a model for probability distributions on image spaces. We show that any distribution of images can be factored exactly into conditional distributions of feature vectors at one resolution (pyramid level) conditioned on the image information at lower resolutions. We would like to factor this over positions in the pyramid levels to make it tractable, but such factoring may miss long-range dependencies. To capture long-range dependencies, we introduce hidden class labels at each pixel in the pyramid. The result is a hierarchical mixture of conditional probabilities, similar to a hidden Markov model on a tree. The model parameters can be found with maximum likelihood estimation using the EM algorithm. We have obtained encouraging preliminary results on the problems of detecting various objects in SAR images and target recognition in optical aerial images.

1. INTRODUCTION

Many approaches to object recognition in images estimate $\Pr(C \mid I)$, the probability that an object of class C is present in an image I. By contrast, a model of the probability distribution of images, $\Pr(I \mid C)$, has many attractive features. We could use this for object recognition in the usual way by training a distribution for each object class and using Bayes' rule to get $\Pr(C \mid I)$, or by using the likelihood ratio between $\Pr(I \mid C)$ and $\Pr(I \mid \bar{C})$. Clearly there are many other uses for image distributions, since any kind of data analysis task can be approached using knowledge of the distribution of the data. For classification we could attempt to detect unusual examples and reject them, rather than trusting the classifier's output. We could also compress, segment, interpolate, suppress noise, extend resolution, fuse multiple images, etc.

Many image analysis algorithms use probability concepts, but few treat the distribution of images, e.g., maximum entropy modeling [1]. There are several approaches that do not model the probability distribution on an image space, but motivated our work, e.g., MRF models [2, 3], the flexible histogram approach [4, 5], and multiscale stochastic processes [6]. All of these methods seem to be well-suited for modeling texture, but it is unclear how we might use them to capture the appearance of more structured objects.

As in many other approaches, we model the distribution of local image structure by using some local features, namely the outputs of some filters, and capture longer-range (either in scale or position) dependencies by modeling the influence of neighboring structures on each other. However, we argue that the presence of objects in images can make local conditioning like this inadequate. We capture these long-range dependencies by using hidden variables. The dependencies between the hidden variables in our model are local, like those in some MRF models, but marginalizing over them introduces long-range dependencies. We expect that such hidden variables would give poor models of object structure if they were only implemented at one pyramid level. Therefore we introduce them at all levels in a pyramid, and give them coarse to fine dependence.

2. THE HIP MODEL

To show that such a model can be a proper distribution on an image space, we show that any distribution on an image space can be factored into a coarse to fine hierarchy of conditional distributions. From an image I we build a Gaussian pyramid. Call the l-th level I_l, e.g., the original image is I_0. From each Gaussian level I_l we extract some set of feature images $\mathbf{F}_l$ (Figure 1). Sub-sample these to get feature images $\mathbf{G}_l$, so that the images in $\mathbf{G}_l$ have the same dimensions as I_{l+1}. Denote the set of images $\{I_{l+1}, \mathbf{G}_l\}$ by $\tilde{\mathbf{G}}_l$, and the mapping from I_l to $\tilde{\mathbf{G}}_l$ by $\tilde{\mathcal{G}}_l$. If $\tilde{\mathcal{G}}_l$ is invertible for all $l \in \{0, \ldots, L-1\}$ it is easy to show that

$$\Pr(I) = \left[\prod_{l=0}^{L-1} |\tilde{\mathcal{G}}_l| \Pr(\mathbf{G}_l \mid I_{l+1})\right] \Pr(I_L) \qquad (1)$$

In order to factor $\Pr(\mathbf{G}_l \mid I_{l+1})$ over positions, we introduce hidden variables. There is enormous freedom in this choice, although different choices can be easier or harder

We thank Jeremy De Bonet and John Fisher for kindly answering questions about their work and experiments. This work supported by the United States Government.

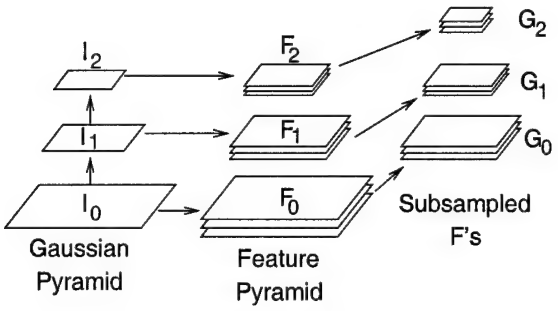

Fig. 1. Pyramids and feature notation.

to work with. One simple but non-trivial choice is to introduce an image A_l of integers at each level l. We assume that these contain enough information to allow us to factor $\Pr(\mathbf{G}_l \mid I_{l+1})$. Furthermore we assume that the local hidden variable $a_l(x)$ and the local lower-resolution feature vector $\mathbf{f}_{l+1}(x)$ carry all of the information in I_{l+1} that is relevant to the local feature vector $\mathbf{g}_l(x)$. This gives

$$\Pr(I) \propto \sum_{A_0,\ldots,A_{L-1}} \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Big[\Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l, x) \times \Pr(a_l \mid a_{l+1}, x)\Big], \quad (2)$$

where $a_{l+1}(x)$ is the hidden variable at the *parent* of x in the tree structure given by the sub-sampling operation. (To avoid repeating the string "(x)", we specify the location x as a conditioning variable in each $\Pr()$.)

3. TRAINING WITH EM

This model can be fit to data using an Expectation-Maximization (EM) algorithm. The E-step is the sum over hidden variables, which is tractable thanks to the tree structure of their dependencies. We choose $\Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l)$ to be normal with a mean that depends linearly on $\mathbf{f}_{l+1}$, i.e., $\Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l) = \mathcal{N}(M_{a_l}\mathbf{f}_{l+1} + \bar{\mathbf{g}}_{a_l}, \Lambda_{a_l})$. This makes the M-step tractable, and is rich enough to reproduce the non-Gaussian distribution of neighboring features on each other (see [7]). To enforce normalization we parameterize the label probabilities as $\Pr(a_l \mid a_{l+1}) = \pi_{a_l,a_{l+1}} / \sum_{a_l} \pi_{a_l,a_{l+1}}$. We denote by $\theta = \{\bar{\mathbf{g}}_{a_l}, M_{a_l}, \Lambda_{a_l}, \pi_{a_l,a_{l+1}}, \forall a_l, \forall l\}$ the vector of all parameters. For brevity we simply reproduce the relevant formulas without derivations.

To compute the expectations in the EM algorithm we need the joint probabilities of the image and individual labels at a position and pyramid level. These are given as

$$\Pr(a_l, a_{l+1}, x, I \mid \theta^t) = u_l(a_l, x)\tilde{d}_l(a_{l+1}, x)\Pr(a_l|a_{l+1}) \quad (3)$$

$$\Pr(a_l, x, I \mid \theta^t) = u_l(a_l, x)d_l(a_l, x), \quad (4)$$

where θ^t is the parameter vector from the t-th EM iteration. The quantities u and d are obtained through the upward and downward recursion relations

$$u_l(a_l, x) = \Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l, x) \prod_{x' \in \mathsf{Ch}(x)} \tilde{u}_{l-1}(a_l, x') \quad (5)$$

$$\tilde{u}_l(a_{l+1}, x) = \sum_{a_l} \Pr(a_l \mid a_{l+1}) u_l(a_l, x) \quad (6)$$

$$d_l(a_l, x) = \sum_{a_{l+1}} \Pr(a_l \mid a_{l+1}) \tilde{d}_l(a_{l+1}, x) \quad (7)$$

$$\tilde{d}_l(a_{l+1}, x) = \frac{u_{l+1}(a_{l+1}, \mathsf{Par}(x))}{\tilde{u}_l(a_{l+1}, x)} d_{l+1}(a_{l+1}, \mathsf{Par}(x)). \quad (8)$$

Here $\mathsf{Ch}(x)$ is the set of pixel locations in some level l that are children of pixel x in level $l+1$ in a tree relationship of pixels in the pyramid. Similarly, $\mathsf{Par}(x)$ is the parent pixel of x.

The upward recursion relations (5 – 6) is initialized at $l = 0$ with $u_0(a_0, x) = \Pr(\mathbf{g} \mid \mathbf{f}_1, a_0, x)$ and ends at $l = L$. At layer L (6) reduces to $\tilde{u}_L(a_{L+1}, x) = \tilde{u}_L(x)$.[1] Since we do not model any further dependencies beyond layer L, the pixels at layer L are assumed independent. The product of all $\tilde{u}_L(x)$ coincides with the total image probability, $\Pr(I|\theta^t) = \prod_{x \in I_L} \tilde{u}_L(x) = u_{L+1}$. The downward recursion (7 – 8) can be executed, starting with equation (8) at $l = L$ with $d_{L+1}(a_{L+1}, x) = d_{L+1}(x) = 1$.[1]

For the update equations, let us denote the average over position at level l weighted by $\Pr(a_l, x \mid I, \theta^t)$ by $\langle . \rangle_{t,a_l}$, i.e.,

$$\langle X \rangle_{t,a_l} = \frac{\sum_x \Pr(a_l, x \mid I, \theta^t) X(x)}{\sum_x \Pr(a_l, x \mid I, \theta^t)}. \quad (9)$$

Then the update equations for the Gaussian parameters are

$$M_{a_l}^{t+1} = \left(\langle \mathbf{g}_l \mathbf{f}_{l+1}^T \rangle_{t,a_l} - \langle \mathbf{g}_l \rangle_{t,a_l} \langle \mathbf{f}_{l+1}^T \rangle_{t,a_l}\right) \times \left(\langle \mathbf{f}_{l+1} \mathbf{f}_{l+1}^T \rangle_{t,a_l} - \langle \mathbf{f}_{l+1} \rangle_{t,a_l} \langle \mathbf{f}_{l+1}^T \rangle_{t,a_l}\right)^{-1}, \quad (10)$$

$$\bar{\mathbf{g}}_{a_l}^{t+1} = \langle \mathbf{g}_l \rangle_{t,a_l} - M_{a_l}^{t+1} \langle \mathbf{f}_{l+1} \rangle_{t,a_l}, \quad (11)$$

and

$$\Lambda_{a_l}^{t+1} = \left\langle \left(\mathbf{g}_l - M_{a_l}^{t+1}\mathbf{f}_{l+1}\right)\left(\mathbf{g}_l - M_{a_l}^{t+1}\mathbf{f}_{l+1}\right)^T \right\rangle_{t,a_l} - \bar{\mathbf{g}}_{a_l}^{t+1} \bar{\mathbf{g}}_{a_l}^{t+1\,T}. \quad (12)$$

The update equation for the label probability parameters is

$$\pi_{a_l,a_{l+1}}^{t+1} = \sum_x \Pr(a_l, a_{l+1}, x | I, \theta^t). \quad (13)$$

[1] The (non-existent) label a_{L+1} can be thought of as a label with a single possible value, which is always set. The conditional $\Pr(a_L \mid a_{L+1})$ turns then into a prior $\Pr(a_L)$

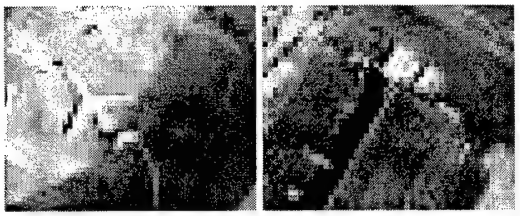

Fig. 2. Examples of positive (left) and negative (right) ROIs for the aircraft detection problem. Data from the MassGIS at `http://ortho.mit.edu/nsdi/`.

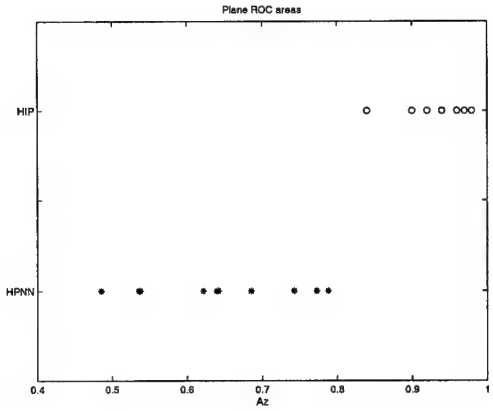

Fig. 3. A_z values from a jack-knife study of detection performance of HIP and HPNN (hybrid pyramid/neural network) models.

4. EXPERIMENTS

We have applied this HIP model to two problems. The first was to detect aircraft in aerial photographs. The HIP model performed substantially better than our own hybrid pyramid neural network (HPNN) algorithm [8]. (See Figures 2 and 3.) (For a better comparison we would select features independently for the HIP and HPNN models. The HPNN gave $A_z = 0.86$ with a different set of features.)

For vehicle discrimination in SAR, we performed an experiment with the three target classes in the MSTAR public targets data set, to compare with the results of the flexible histogram approach of De Bonet, et al [5]. We trained three HIP models, one for each of the target vehicles BMP-2, BTR-70 and T-72 (Figure 4). As in [5] we trained each model on ten images of its class, one image for each of ten aspect angles, spaced approximately 36° apart. We trained one model for all ten images of a target, whereas De Bonet et al trained one model per image.

We first discriminated between vehicles of one class and other objects by thresholding $\log \Pr(I \mid C)$, i.e., no model of other objects is used. For the tests, the other objects were taken from the test data for the two other vehicle classes,

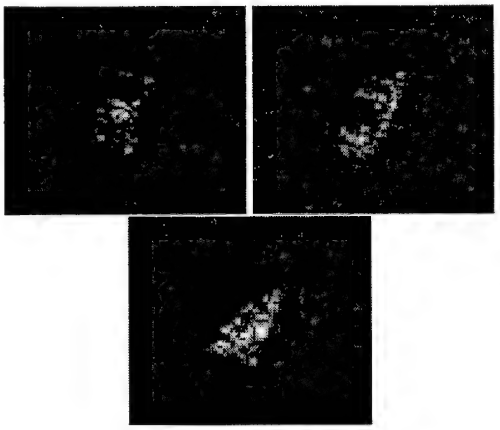

Fig. 4. SAR images of three vehicle classes. Data from the MSTAR public data set.

plus seven other vehicle classes. There were 1,838 image from these seven other classes, 391 BMP2 test images, 196 BTR70 test images, and 386 T72 test images. The resulting ROC curves are shown in Figure 5a.

A second discrimination criterion that uses a distribution is the likelihood ratio, $\log \Pr(I \mid C_1) - \log \Pr(I \mid C_2)$. Here we cannot use the extra seven vehicle classes. The resulting ROC curves are shown in Figure 5b. The performance is comparable to that of the flexible histogram approach of De Bonet et al.

5. CONCLUSION

We have presented a hierarchical image probability (HIP) model for probability distributions of images, and demonstrated its utility in a pair of object recognition tasks. The model uses hidden class labels to capture long-range dependencies. A distribution model has many potential uses besides recognition, including compression, noise suppression, novelty detection, segmentation, etc.

The HIP model has two key elements. First is the restriction that the features be invertible to make the model a proper probability distribution on the image space. It appears to be possible to relax these restrictions in some cases. Second is the use of hidden variables, since these are needed to express long-range dependencies in the model. Our current hidden variable structure was chosen for tractability, since we can explicitly marginalize the hidden variables in this structure. Generalizations like choosing a connectivity denser than a tree, or including continuous hidden variables could have benefits, but we would need approximations to evaluate the probabilities. There is much room for further work along these lines.

We are also working on sampling from HIP models, i.e., generating random images. This capability provides an in-

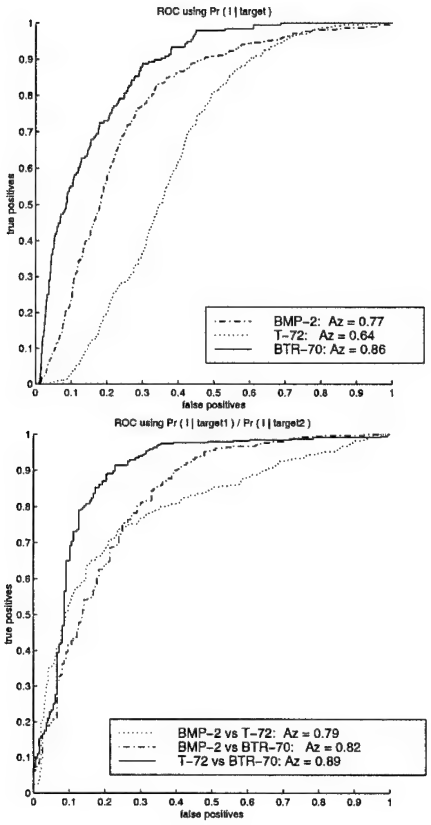

Fig. 5. ROC curves for vehicle detection in SAR imagery. (Upper: ROC curves by thresholding HIP likelihood of desired class. Lower: ROC curves for inter-class discrimination using ratios of likelihoods as given by HIP models.

dependent means of evaluating the model that is not available with neural network models of $\Pr(C \mid I)$.

6. REFERENCES

[1] Song Chun Zhu, Ying Nian Wu, and David Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Computation*, vol. 9, no. 8, pp. 1627–1660, 1997.

[2] Stuart Geman and Donald Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. PAMI*, vol. PAMI-6, no. 6, pp. 194–207, Nov. 1984.

[3] Rama Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. ASSP*, vol. 33, pp. 959–963, 1985.

[4] Jeremy S. De Bonet and Paul Viola, "Texture recognition using a non-parametric multi-scale statistical model," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 1998.

[5] J. S. De Bonet, P. Viola, and J. W. Fisher III, "Flexible histograms: A multiresolution target discrimination model," in *Proceedings of SPIE*, E. G. Zelnio, Ed., 1998, vol. 3370.

[6] Mark R. Luettgen and Alan S. Willsky, "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination," *IEEE Trans. Image Proc.*, vol. 4, no. 2, pp. 194–207, 1995.

[7] Robert W. Buccigrossi and Eero P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," Tech. Rep. 414, U. Penn. GRASP Laboratory, 1998, Available at ftp://ftp.cis.upenn.edu/pub/eero/buccigrossi97.ps.gz.

[8] Clay D. Spence and Paul Sajda, "Applications of multiresolution neural networks to mammography," in *Advances in Neural Information Processing Systems 11*, Michael S. Kearns, Sara A. Solla, and David A. Cohn, Eds., Cambridge, MA, 1998, pp. 981–988, MIT Press.

Detection, Synthesis and Compression in Mammographic Image Analysis with a Hierarchical Image Probability Model

Clay Spence Lucas Parra Paul Sajda

Vision Technologies, Sarnoff Corporation, Princeton, NJ 08540
Vision Technologies, Sarnoff Corporation, Princeton, NJ 08540
Biomedical Engineering, Columbia University, New York, NY 10023

Abstract

We develop a probability model over image spaces and demonstrate its broad utility in mammographic image analysis. The model employs a pyramid representation to factor images across scale and a tree-structured set of hidden variables to capture long-range spatial dependencies. This factoring makes the computation of the density functions local and tractable. The result is a hierarchical mixture of conditional probabilities, similar to a hidden Markov model on a tree. The model parameters are found with maximum likelihood estimation using the EM algorithm. The utility of the model is demonstrated for three applications; 1) detection of mammographic masses in computer-aided diagnosis 2) qualitative assessment of model structure through mammographic synthesis and 3) lossless compression of mammographic regions of interest.

1. Introduction

In mammographic computer-assisted diagnosis (CAD) one typically estimates $\Pr(C|I)$, the conditional probability of class C (e.g. benign vs. malignant) given image I or a set of features extracted from I. Previous efforts have concentrated on the development of such *discriminant* models for CAD [1][2][3][4][?]. By contrast, a *generative* model, $\Pr(I|C)$, has many attractive features. Classification is possible by training a distribution for each class and using Bayes' rule to obtain $\Pr(C|I) = \Pr(I|C)\Pr(C)/\Pr(I)$. However there are many other benefits of having a model of the distribution of images, since any type of image analysis can be approached using knowledge of the distribution of the data. For example, anomalous images can be detected and rejected, rather than trusting the classifier's output. A generative model can also be used to compress, interpolate, suppress noise, increase or extend resolution, and fuse multiple images.

In the computer vision and pattern recognition community there has been limited work directed at developing probabilities for images. One of the few examples of image distribution models is that constructed by Zhu, Wu and Mumford[5]. In their approach they compute the maximum entropy distribution given a set of statistics across a number of features. Though this approach works well for textures, it is not clear how well it will model the appearance of more structured objects. Several algorithms have investigated modeling the distributions of features extracted from the image, instead of the image itself. The Markov Random Field (*MRF*) models are one such example; see, e.g., References [6, 7]. However, these models tend to be computationally expensive.

Recently, De Bonet and Viola's proposed a flexible histogram approach[8, 9], where features are extracted at multiple image scales, with the resulting feature vectors treated as a set of independent samples drawn from a distribution. The distribution of feature vectors is then modeled using Parzen windows. Though they report good results, their model treats the feature vectors from neighboring pixels as independent samples when in fact they share exactly the same components from lower-resolutions. One solution to this is to build a model in which the features at one pixel of one pyramid level condition the features at each of several child pixels at the next higher-resolution pyramid level. The multiscale stochastic process (*MSP*) methods do exactly that. Luettgen and Willsky[10], for example, applied a scale-space auto-regression (AR) model to texture discrimination. They use a quadtree or quadtree-like organization of the pixels in an image pyramid, and model the features in the pyramid as a stochastic process from coarse-to-fine levels along the tree. The variables in the process are hidden, and the observations are sums of these hidden variables plus noise. However the assumed Gaussian distributions are a limitation of MSP models as well as the fact that the model is of the probability of the observations on the tree, not of the image.

All of these methods appear well-suited for modeling texture, but it is unclear how one might build models to capture the appearance of more structured objects. For

example, in mammography, benign and malignant masses tend to be characterized by a combination of texture and shape features[11] and may also include contextual influences. Therefore local conditioning, like that of the flexible histogram and MSP approaches, is inadequate.

Recently, several groups have developed what are essentially extensions of the MSP models by adding hidden variables. These can be seen as improving the model's ability to capture non-local dependencies in the image. For example, Crouse et al developed their Hidden Markov Tree (*HMT*) models [12] for signals and images. A primary motivation of these models is to capture the tendency for wavelet coefficients to group into two classes, one with large and the other with small coefficent magnitudes. Thus their hidden states have one of two values corresponding to large and small wavelet coefficients. This is well suited to the many signal and image types that have homogeneous regions with boundaries. These models have been successfully applied to several problems, especially image denoising and texture segmentation. Cheng and Bouman [13] applied another model of this sort for segmentation, in which the observed class labels play the role of hidden variables, and so of course are no longer hidden.

We have independently developed a class of models for probability distributions of images that we call hierarchical image probability (*HIP*) models. These also have tree-structured graph of the dependencies between hidden variables at different scales, and use mixtures of multivariate Gaussians to model the local distributions of vectors of features. In the following we present the basic HIP models, along with EM algorithm for training the models. We show preliminary results of the application of HIP models to mammographic image analysis, including lesion classification, mammographic synthesis and compression of mammographic ROIs.

2 Coarse-To-Fine Factoring Of Image Distributions

Our goal will be to write the image distribution in a form similar to $\Pr(I) \sim \Pr(\mathbf{F}_0 \,|\, \mathbf{F}_1) \Pr(\mathbf{F}_1 \,|\, \mathbf{F}_2) \ldots$, where $\mathbf{F}_l$ is the set of feature images at pyramid level l. We expect that the short-range dependencies can be captured by the model's distribution of individual feature vectors, while the long-range dependencies can be captured at low resolution. The large-scale structures affect finer scales by the conditioning.

We first prove that a coarse-to-fine factoring like this is correct. From an image I we build a Gaussian pyramid (repeatedly blur-and-subsample, with a Gaussian filter). Call the l-th level I_l, e.g., the original image is I_0 (Figure 1). From each Gaussian level I_l we extract a set of feature images $\mathbf{F}_l$. Sub-sample these to get feature images $\mathbf{G}_l$. Note that the images in $\mathbf{G}_l$ have the same dimensions as I_{l+1}. We denote by $\tilde{\mathbf{G}}_l$ the set of images containing I_{l+1} and the images in $\mathbf{G}_l$. We further denote the mapping from I_l to $\tilde{\mathbf{G}}_l$ by $\tilde{\mathcal{G}}_l$.

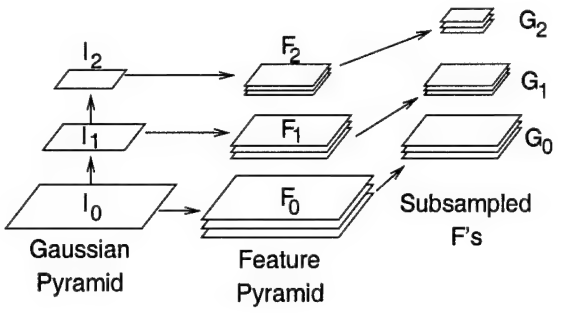

Figure 1: Pyramids and feature notation.

Suppose that $\tilde{\mathcal{G}}_0 : I_0 \mapsto \tilde{\mathbf{G}}_0$ is invertible. Then we can think of $\tilde{\mathcal{G}}_0$ as a change of variables. If we have a distribution on a space, its expressions in two different coordinate systems are related by multiplying by the Jacobian. In this case we get $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\tilde{\mathbf{G}}_0)$. Since $\tilde{\mathbf{G}}_0 = (\mathbf{G}_0, I_1)$, we can factor $\Pr(\tilde{\mathbf{G}}_0)$ to get $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\mathbf{G}_0 \,|\, I_1) \Pr(I_1)$. If $\tilde{\mathcal{G}}_l$ is invertible for all $l \in \{0, \ldots, L-1\}$ then we can simply repeat this change of variable and factoring procedure to get

$$\Pr(I) = \left[\prod_{l=0}^{L-1} |\tilde{\mathcal{G}}_l| \Pr(\mathbf{G}_l \,|\, I_{l+1}) \right] \Pr(I_L) \qquad (1)$$

This is a very general result, valid for all $\Pr(I)$, with some rather weak restrictions to make the change of variables valid. The restriction that $\tilde{\mathcal{G}}_l$ be invertible is strong, but many such feature sets are known to exist, e.g., most wavelet transforms on images.

3 The Need For Hidden Variables

For the sake of tractability we want to factor $\Pr(\mathbf{G}_l \,|\, I_{l+1})$ over positions, for example

$$\Pr(I) \sim \prod_l \prod_{x \in I_{l+1}} \Pr\big(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x)\big)$$

where $\mathbf{g}_l(x)$ and $\mathbf{f}_{l+1}(x)$ are the feature vectors at position x. The dependence of $\mathbf{g}_l$ on $\mathbf{f}_{l+1}$ expresses the persistence of image structures across scale, e.g., an edge is usually detectable as such in several neighboring pyramid levels. The flexible histogram and MSP methods share this structure.

While it may be plausible that $\mathbf{f}_{l+1}(x)$ has a strong influence on $\mathbf{g}_l(x)$, a model distribution with this factorization

and conditioning cannot capture some properties of real images. Objects in the world cause correlations and non-local dependencies in images. For example, the presence of a particular object might cause a certain kind of texture to be visible at level l. Usually local features $\mathbf{f}_{l+1}$ by themselves will not contain enough information to infer the object's presence, but the entire image I_{l+1} at that layer might. Thus $\mathbf{g}_l(x)$ is influenced by more of I_{l+1} than the local feature vector.

Similarly, objects create long-range dependencies. For example, an object class might result in a specific kind of texture across a large area of the image (e.g. malignant breast masses tend to have inhomogenous region enhancement). If an object of this class is always present, the distribution may factor, but if such objects are not always present and cannot be inferred from lower-resolution information, the presence of the texture at one location affects the probability of its presence elsewhere.

To capture these long-range dependencies we introduce hidden variables to represent the non-local information that is not captured by local features. These hidden varibles also constrain the variability of features at the next finer scale. Denoting the hidden variables collectively by A, we assume that conditioning on A allows the distributions over feature vectors to factor. In general, the distribution over images becomes

$$\Pr(I) \propto \sum_A \Big\{ \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Pr(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x), A) \times \Pr(A \,|\, I_{L+1}) \Big\} \Pr(I_{L+1}). \quad (2)$$

This is a very general form for A and we instead would like to be more specific. In particular we would like to preserve the conditioning of higher-resolution information on coarser-resolution information, and the ability to factor over positions. This lead to the following structure for our HIP model:[1]

$$\Pr(I) \propto \sum_{A_0,\ldots,A_L} \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Big[\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \times \Pr(a_l \,|\, a_{l+1}, x) \Big] \quad (3)$$

To each position x at each level l we attach a hidden discrete index or label $a_l(x)$. The resulting label image A_l for level l has the same dimensions as the images in $\tilde{\mathbf{G}}_l$.

[1] In principle there is also a factor of $\Pr(I_{L+1})$. In many cases I_{L+1} will be a single pixel that is approximately the mean brightness in the image. We ignore this, which is equivalent to assuming that $\Pr(I_{L+1})$ is flat over some range. In this case $\mathbf{f}_{L+1}$ is zero for typical features. In addition, there is no hidden variable a_{L+1}. If we combine these considerations we see that the $l = L$ factor should be read as $\prod_x \Pr(\mathbf{g}_L \,|\, a_L, x) \Pr(a_L, x)$.

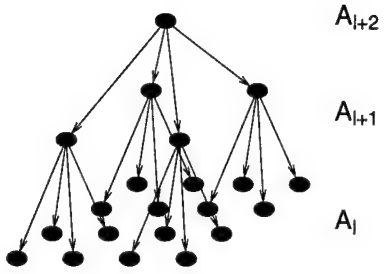

Figure 2: Quadtree structure of the conditional dependency between hidden variables in the HIP model.

Since $a_l(x)$ codes non-local information we can think of the labels A_l as a learned segmentation at the l-th pyramid level. By conditioning $a_l(x)$ on $a_{l+1}(x)$, we mean that $a_l(x)$ is conditioned on a_{l+1} at the *parent* pixel of x. This parent-child relationship follows from the sub-sampling operation. For example, if we sub-sample by two in each direction to get $\mathbf{G}_l$ from $\mathbf{F}_l$, we condition the variable a_l at (x, y) in level l on a_{l+1} at location $(\lfloor x/2 \rfloor, \lfloor y/2 \rfloor)$ in level $l+1$ (Figure 2). This gives a tree structure to the dependency graph of the hidden variables, i.e. a belief network. By conditioning child labels on their parents information propagates though the layers to other areas of the image while accumulating information along the way.

For simplicity we have chosen $\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l)$ to be normal with a mean that depends linearly on $\mathbf{f}_{l+1}$,

$$\Pr(\mathbf{g} \,|\, \mathbf{f}, a) = \mathcal{N}(\mathbf{g}, M_a \mathbf{f} + \bar{\mathbf{g}}_a, \Lambda_a) \quad (4)$$

4 EM Algorithm

Due to the tree structure, the belief network for the hidden variables is relatively straightforward to train with an EM algorithm. The expectation step (summing over a_l's) can be performed directly.[2] The expectation is weighted by the probability of a label or a parent-child pair of labels given the image. This can be computed in a fine-to-coarse-to-fine procedure, i.e. working from leaves to the root and then back out to the leaves. The method is based on belief propagation [14].

Once the expectations are computed, the normal distribution makes the M-step tractable; one simply computes the updated $\bar{\mathbf{g}}_{a_l}$, Σ_{a_l}, M_{a_l}, and $\Pr(a_l \,|\, a_{l+1})$ as combinations of various expectation values.

In order to apply the EM algorithm, a parameterization for the model is required. The parameterization of $\Pr(\mathbf{g} \,|\, \mathbf{f}, a)$ is given above in Equation 4. For $\Pr(a_l \,|\, a_{l+1})$

[2] Note that a more densely-connected structure, with each child having several parents, we have required either an approximate algorithm or Monte Carlo techniques.

we use the parameterization

$$\Pr(a_l \mid a_{l+1}) = \frac{\pi_{a_l,a_{l+1}}}{\sum_{a_l} \pi_{a_l,a_{l+1}}} \tag{5}$$

in order to ensure proper normalization.

Below, we denote the new parameter values computed during the t-th maximization step as θ^{t+1} and the old values as θ^t.

4.1 Maximization

Maximizing the expectation of the likelihood over the hidden variables with respect to the model parameters gives the following update formulae:

$$\pi^{t+1}_{a_l,a_{l+1}} = \sum_x \Pr(a_l, a_{l+1}, x | I, \theta^t), \tag{6}$$

$$M^{t+1}_{a_l} = \left(\langle \mathbf{g}_l \mathbf{f}^T_{l+1} \rangle_{t,a_l} - \langle \mathbf{g}_l \rangle_{t,a_l} \langle \mathbf{f}^T_{l+1} \rangle_{t,a_l}\right) \times \left(\langle \mathbf{f}_{l+1} \mathbf{f}^T_{l+1} \rangle_{t,a_l} - \langle \mathbf{f}_{l+1} \rangle_{t,a_l} \langle \mathbf{f}^T_{l+1} \rangle_{t,a_l}\right)^{-1}, \tag{7}$$

$$\bar{\mathbf{g}}^{t+1}_{a_l} = \langle \mathbf{g}_l \rangle_{t,a_l} - M^{t+1}_{a_l} \langle \mathbf{f}_{l+1} \rangle_{t,a_l}, \tag{8}$$

and

$$\Lambda^{t+1}_{a_l} = \left\langle \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1}\right) \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1}\right)^T \right\rangle_{t,a_l} - \bar{\mathbf{g}}^{t+1}_{a_l} \bar{\mathbf{g}}^{t+1\,T}_{a_l}. \tag{9}$$

Here the brackets $\langle . \rangle_{t,a_l}$ denote the expectation value

$$\langle X \rangle_{t,a_l} = \frac{\sum_x \Pr(a_l, x \mid I, \theta^t) X(x)}{\sum_x \Pr(a_l, x \mid I, \theta^t)}. \tag{10}$$

4.2 Expectation

In the E-step we need to compute the probabilities of pairs of labels from neighboring layers $\Pr(a_l, a_{l+1}, x_l \mid I, \theta^t)$ for given image data. Note that in all occurrences of the reestimation equations, i.e. (5,6) and (10), we require that quantity only up to an overall factor. We can choose that factor to be $\Pr(I|\theta^t)$ and can compute $\Pr(a_l, a_{l+1}, x_l, I|\theta^t)$ instead using

$$\Pr(a_l, a_{l+1}, x \mid I, \theta^t) \Pr(I \mid \theta^t) = \Pr(a_l, a_{l+1}, x, I \mid \theta^t) = \sum_{A \setminus a_l(x), a_{l+1}(x)} \Pr(I, A|\theta^t) \tag{11}$$

The computation of these quantities can be cast as recursion formulae, defined in terms of quantities u and d, which approximately represent upwards and downwards propagating probabilities. The recursion formulae are

$$u_l(a_l, x) = \Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l, x) \times \prod_{x' \in \mathsf{Ch}(x)} \tilde{u}_{l-1}(a_l, x') \tag{12}$$

$$\tilde{u}_l(a_{l+1}, x) = \sum_{a_l} \Pr(a_l | a_{l+1}) u_l(a_l, x) \tag{13}$$

$$d_l(a_l, x) = \sum_{a_{l+1}} \Pr(a_l | a_{l+1}) \tilde{d}_l(a_{l+1}, x) \tag{14}$$

$$\tilde{d}_l(a_{l+1}, x) = \frac{u_{l+1}(a_{l+1}, \mathsf{Par}(x))}{\tilde{u}_l(a_{l+1}, x)} \times d_{l+1}(a_{l+1}, \mathsf{Par}(x)) \tag{15}$$

The upward recursion relations (12–13) are initialized at $l = 0$ with $u_0(a_0, x) = \Pr(\mathbf{g} \mid \mathbf{f}_1, a_0, x)$ and end at $l = L$. At level L Equation 13 reduces to $\tilde{u}_L(a_{L+1}, x) = \tilde{u}_L(x)$.[3] Since we do not model any further dependencies beyond layer L, the pixels at layer L are assumed independent. Considering the definition of u, it is evident that the product of all $\tilde{u}_L(x)$ coincides with the total image probability,

$$\Pr(I|\theta^t) = \prod_{x \in I_L} \tilde{u}_L(x) = u_{L+1}. \tag{16}$$

The downward recursion (14 - 15) can be executed, starting with equation (15) at $l = L$ with $d_{L+1}(a_{L+1}, x) = d_{L+1}(x) = 1$.[3] The downwards recursion ends at $l = 0$ with equation (14).

We can now compute (11) as

$$\Pr(a_l, a_{l+1}, x, I \mid \theta^t) = u_l(a_l, x) \tilde{d}_l(a_{l+1}, x) \times \Pr(a_l | a_{l+1}) \tag{17}$$

$$\Pr(a_l, x, I \mid \theta^t) = u_l(a_l, x) d_l(a_l, x) \tag{18}$$

Computations (12–18) in the E-step at iteration t are done with fixed parameters θ^t.

5 Experimental Results

In this section we report some of our preliminary results for applying the HIP model to mammographic image analysis.

5.1 Mass Detection

To demonstrate utility, we use HIP as a post-processor (i.e. adjunct) to the University of Chicago's (UofC) CAD system[15]. False positive and true positive regions of interest (ROIs) were output from the UofC CAD system and

[3] The (non-existent) label a_{L+1} can be thought of as a label with a single possible value, which is always set. The conditional $\Pr(a_L | a_{L+1})$ turns then into a prior $\Pr(a_L)$

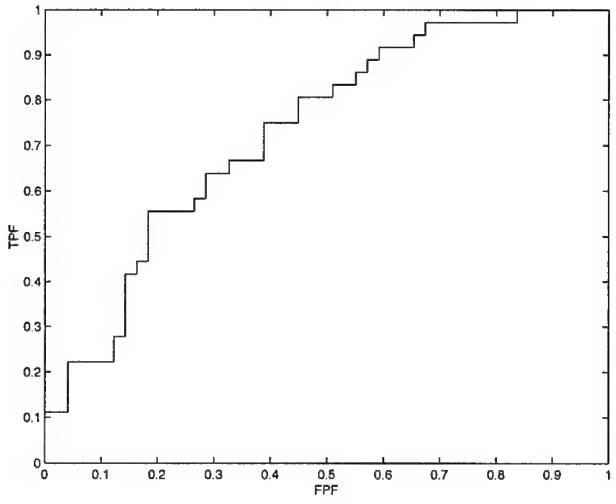

Figure 3: ROC curve for results of HIP models used as a post-processor for mass detection in the University of Chicago's mammographic CAD system.

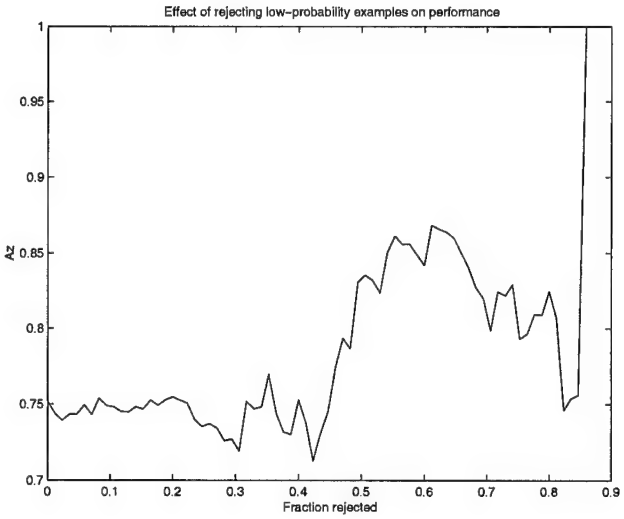

Figure 4: Using the HIP model for novelty detection and generating confidence measures. Thresholding the absolute value of the likelihood (abscissa) enables rejection of a fraction of the data that is novel, relative to the data on which the models were trained. This acts as a confidence measure, which can improve the performance of the model (Az values on ordinate axis).

used for training and testing. The goal was to determine if the HIP model could be used to reduce false positives without significant loss in sensitivity.

Two HIP models were trained; one using 36 randomly-chosen ROIs that contained masses, and a second trained on 48 randomly-chosen ROIs without masses. The likelihood ratio under the two models was used as the test criterion, i.e., a threshold on this ratio is used to decide which ROIs will be detected as masses. The true and false positive rates as a function of the threshold were measured on a novel test set consisting of 36 mass and 49 non-mass ROIs.

A search was performed over the number of hidden labels values at each level. The search criterion used the negative log-likelihood on the training data plus the minimum-description-length penalty term, $d \log(N)/2$, where d is the number of model parameters and N is the the number of training examples [16]. The maximum number of labels in a level was bounded at 17.

The best perfroming model had an architecture of 17, 17, 11, 2, and 1 hidden label in levels 0–4, respectively. The receiver operatin characteristic (ROC) curve [17] for the test images is shown in Figure 3. For this architecture the area under the curve (A_z) was 0.75. For this architecture and set of parameter the HIP model is able to eliminate 17% of the false positives generated by the UofC CAD system, without loss in sensitivity.

5.2 Novelty Detection

Novelty detection identifies examples that are significantly different from the examples on which the model(s) was trained [18]. Detecting novel examples can be useful in a CAD system for generating confidence measures on the CAD output and identifying data that could be used in future training of the model. The HIP model's generative structure enables novel examples to be identified by thresholding the log-likelihood of the models. Figure 4 illustrates how ROC performance improves if novelty detection is used to generate a confidence measure for rejecting low-confidence examples. In this example, two HIP models were trained, one for positive ROIs and one for negatives ROIs (same ROI database as for classification). Test data was evaluated by computing the likelihood ratio of the models as well as the absolute value of the log-likelihoods. The absolute value of the log-likelihoods are thresholded such that low values are considered low confidence and therefore rejected (not classified). As the threshold on the log-likelihood is increased, more ROIs are rejected because of low confidence and the area under the ROC curve increases.

5.3 Mammographic Synthesis

Since the HIP model is a generative model, we can sample the model and synthesize new images. In the context of ROI classification, synthesized images can provide qualitative insight into what features the model is extracting and representing for both positive and negative ROIs. Using the same

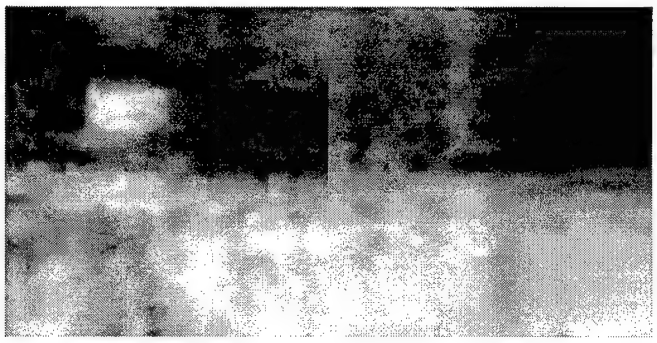

Figure 5: Mammographic ROI images synthesized from positive and negative HIP models. Synthesized positive ROIs (left) tend to have more focal structure, with more defined borders and higher spatial frequency content. Negative ROIs (right) tend to be more amorphous with lower spatial frequency content.

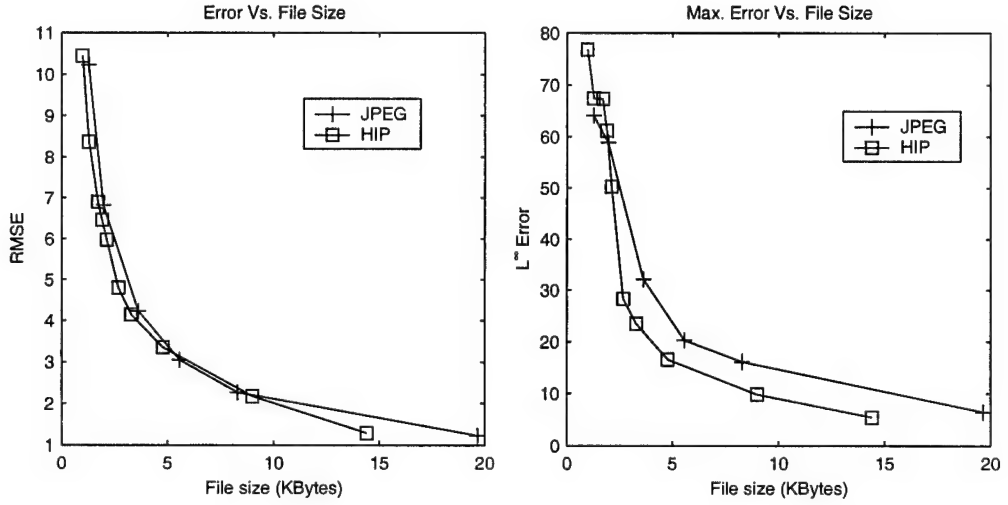

Figure 6: (Left)Root mean-squared error vs. size of compressed file, JPEG and HIP. (Right) Maximum error (L^∞ norm) vs. size of compressed file, JPEG and HIP.

ROI database used for classification, we constructed HIP models for positives (masses) and negatives (no masses). The trained HIP models were sampled to synthesize new ROI images. The sampling procedure begins at the coarsest resolution, where the hidden labels are randomly sampled from the distribution $\Pr(A_L)$. The feature images $\mathbf{G}_L$ are then sampled from $\Pr(\mathbf{G}_L \mid A_L)$. The $\mathbf{G}_L$ are used to construct I_{L-1}, from which the $\mathbf{F}_L$ are constructed. We then sample A_{L-1} from $\Pr(A_{L-1} \mid A_L)$, and then $\mathbf{G}_{L-1}$ from $\Pr(\mathbf{G}_{L-1} \mid \mathbf{F}_L, A_{L-1})$. This is repeated until the finest resolution is reached and I_0 is constructed.

Figure 5 shows examples of these images. Inspection of the synthesized positive ROIs shows more focal structure, with more well-defined borders and higher spatial frequency content than the negative ROIs.

5.4 Mammographic Image Compression

A stream of random variables can be optimally compressed if we know their distribution, and so having a HIP model of a source of images should allow us to compress examples of those images with high efficiency. Here we demonstrate compression with HIP models using a simple technique.

Given an image and a HIP model, we compute the most likely value of each hidden label, $a_l^*(x) = \arg\max_{a_l(x)} \Pr(a_l, x, I \mid \theta^t)$ using Equation 18, and code each feature vector $\mathbf{g}_l(x)$ using $\Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l^*, x)$. The latter is used by decomposing $\mathbf{g}_l(x)$ into its components along the eigenvectors of the covariance of $\Pr(\mathbf{g}_l \mid \mathbf{f}_{l+1}, a_l^*, x)$, $\Sigma_{a_l^*}$, and coding those components with a specified precision using Huffman encoders for the Gaussian distributions with variances given by the eigenvalues of $\Sigma_{a_l^*}$. The resulting bitstream was stored in a file that was susbequently

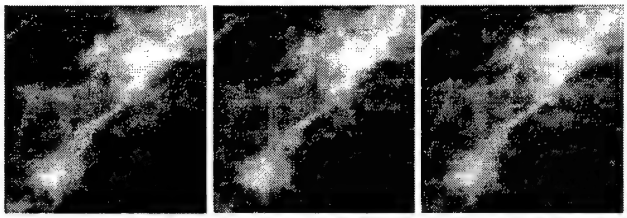

Figure 7: Compression artifacts of JPEG and HIP. Left: Original image, center: JPEG, right: HIP.

compressed with gzip to reduce the redundancy in the many short identical bit patterns. This procedure is currently very computationally expensive, and is not necessarily optimal even if the HIP model exactly matches the image distribution, but it is straightforward to code and serves to demonstrate the capability.

Figure 6 shows the root-mean-squared and maximum errors versus the size of the reulting compressed file, respectively. This is for one randomly-chosen mass ROI image, which was not part of the training set of the HIP model. The HIP algorithm gives mean errors that are comparable to JPEG, and suggests that its maximum errors are a little lower. It is perhaps not surprising, since the HIP model was fit to similar data while JPEG is intended to be general, but it demonstrates the potential. Compressed and uncompressed images are shown in Figure 7.

6 Conclusion

We have developed a class of image probability models we call hierarchical image probability or HIP models. To justify these, we showed that image distributions can be exactly represented as products over pyramid levels of distributions of sub-sampled feature images conditioned on coarser-scale image information. We argued that hidden variables are needed to capture long-range dependencies while allowing us to further factor the distributions over position. In our current model the hidden variables act as indices of mixture components. The resulting model is very similar to the Hidden Markov Tree models, but allows modelling somewhat more general image structures. Because they are models of probability distributions over images, they can be used for a wide range of image processing tasks e.g. classification, compression, noise-suppression, up-sampling, error correction, etc. Here we have presented results for mammographic image analysis. However there are obviously other modalities and medical application areas were HIP models would be useful. One in particular is multi-modal fusion, where the problem is to bring a set of images, acquired using different imaging modalities, into alignment. One method that has demonstrated particularly good performance uses mutual information as an objective criterion [19]. The computation of mutual information requires an estimate of entropies, which in turn requires an estimate of the underlying densities of the images. The HIP model potentially provides a framework for learning those densities.

Acknowledgements

We thank Drs. Robert Nishikawa and Maryellen Giger of The University of Chicago for useful discussions and providing the data. This work was funded by the U.S. Army Medical Research and Material Command (DAMD17-98-1-8061). This paper does not necessarily reflect the position or the policy of the US government, and no official endorsement should be inferred.

References

[1] C.E. Floyd, J.Y. Lo, A.J. Yun, D.C. Sullivan, and P.J. Kornguth, "Prediction of breast cancer malignancy using an artificial neural network," *Cancer*, vol. 74, pp. 2944–2948, 1994.

[2] Y. Jiang, R.M. Nishikawa, D.E. Wolverton, C.E. Metz, M. L. Giger, R.A. Schmidt, and K. Doi, "Automated feature analysis and classification of malignant and benign microcalcifications," *Radiology*, vol. 198, pp. 671–678, 1996.

[3] W. Zhang, K. Doi, M. L. Giger, Y. Wu, R. M. Nishikawa, and R. Schmidt, "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network," *Medical Physics*, vol. 21, no. 4, pp. 517–524, 1994.

[4] S.C. Lo, H.P. Chan, J.S. Lin, H. Li, M.T. Freedman, and S.K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural Networks*, vol. 8, no. (7/8), pp. 1201–1214, 1995.

[5] C. D. Spence and P. Sajda, "Applications of multi-resolution neural networks to mammography," in *Advances in Neural Information Processing Systems 11*,

Michael S. Kearns, Sara A. Solla, and David A. Cohn, Eds., Massachusetts Institute of Technology, Cambridge, MA 02142, 1999, pp. 938–944, MIT Press.

[6] S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Computation*, vol. 9, no. 8, pp. 1627–1660, 1997.

[7] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. PAMI*, vol. PAMI-6, no. 6, pp. 194–207, Nov. 1984.

[8] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. ASSP*, vol. 33, pp. 959–963, 1985.

[9] J. S. De Bonet and P. Viola, "Texture recognition using a non-parametric multi-scale statistical model," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 1998.

[10] J. S. De Bonet, P. Viola, and J. W. Fisher III, "Flexible histograms: A multiresolution target discrimination model," in *Proceedings of SPIE*, E. G. Zelnio, Ed., 1998, vol. 3370.

[11] M. R. Luettgen and A. S. Willsky, "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination," *IEEE Trans. Image Proc.*, vol. 4, no. 2, pp. 194–207, 1995.

[12] D. Kopans, *Breast Imaging*, Lippincott, Philadelphia, PA, 1989.

[13] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.

[14] H. Cheng and C. A. Bouman, "Multiscale bayesian segmentation using a trainable context model," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 511–525, Apr. 2001.

[15] M. I. Jordan, Ed., *Learning in Graphical Models*, vol. 89 of *NATO Science Series D: Behavioral and Brain Sciences*, Kluwer Academic, 1998.

[16] R.M. Nishikawa, R.A. Schmidt, R.B. Osnis, M.L. Giger, K. Doi, and D.E. Wolverton, "Two-year evaluation of a prototype clinical mammographic workstation for computer-aided diagnosis," *Radiology*, vol. 201, no. (P), pp. 256, 1996.

[17] J.A. Rissanen, "Information theory and neural nets," in *Mathematical Perspectives on Neural Networks*, Smolensky, Mozer, and Rumelhart, Eds., 1996, pp. 567–602.

[18] C. Metz, "Current problems in ROC analysis," in *Proceedings of the Chest Imaging Conference*, Madison, WI, Nov. 1988, pp. 315–33.

[19] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, 1995.

[20] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Medical Image Analysis*, vol. 1, no. 1, 1996.

Learning Contextual Relationships in Mammograms using a Hierarchical Pyramid Neural Network

Paul Sajda and Clay Spence

This research was supported under ONR contract N00014-93-C-0202, U.S. Department of the Army contract DAMD17-98-1-8061, the Office of Women's Health DHHS contract No. 282-96-0026, the Murray Foundation and the National Information Display Laboratory.

P. Sajda is with the Department of Biomedical Engineering, Columbia University, New York NY, 10027. He was previously with the Adaptive Image and Signal Processing Group, Sarnoff Research Center, Princeton NJ 08540.

C. Spence is with the Adaptive Image and Signal Processing Group, Sarnoff Research Center, Princeton NJ 08540.

Abstract

This paper describes a pattern recognition architecture, which we term *hierarchical pyramid/neural-network (HPNN)*, that learns to exploit image structure at multiple-resolutions for detecting clinically significant features in digital/digitized mammograms. The HPNN architecture consists of a hierarchy of neural networks, each network receiving feature inputs at a given scale as well as features constructed by networks lower in the hierarchy. Networks are trained using a novel error function for the supervised learning of image search/detection tasks when the position of the objects to be found is uncertain or ill-defined. We have evaluated the HPNN's ability to eliminate false positive regions of interest generated by the University of Chicago's (UofC) Computer-aided Diagnosis (CAD) systems for microcalcification and mass detection. Results show that the HPNN architecture, trained using the UOP error function, reduces the false positive rate of a mammographic CAD system by approximately 50% without significant loss in sensitivity. Investigation into the types of false positives that the HPNN eliminates suggests that the pattern recognizer is automatically learning and exploiting contextual information. Clinical utility is demonstrated through the evaluation of an integrated system in a clinical reader study. We conclude that the HPNN architecture learns contextual relationships between features at multiple scales and integrates these features for detecting microcalcifications and breast masses.

Keywords: mammography, computer-aided diagnosis, hierarchical pyramid neural network, context

I. Introduction

Computer-aided diagnosis (CAD) can be defined as a diagnosis made by a radiologist who incorporates the results of computer analyses of the radiographs [1]. The goal of CAD is to improve radiologists' performance by indicating the sites of potential abnormalities, to reduce the number of missed lesions, and/or by providing quantitative analysis of specific regions in an image to improve diagnosis. CAD systems typically operate as automated "second-opinion" or "double-reading" systems that indicate lesion location and/or type. Since individual human observers overlook different findings, it has been shown that "double reading" (the review of a study by more than one observer) increases the detection rate of breast cancers by 5–15% [2][3][4]. Double reading, if not done efficiently, can significantly increase the cost of screening. Methods to provide improved detection with little increase in costs will have significant impact on the benefits of screening. Automated CAD systems are a promising approach for low-cost double-reading.

Several CAD systems have been in development and the first has been recently approved by the FDA [5]. Complete systems have been rigorously characterized, both in retrospective and prospective trials [6]. Though many have demonstrated clinical utility, there is still a need to reduce false positive rates generated by CAD systems. For example, prospective clinical studies have shown lower sensitivities and specificities than originally found in retrospective studies — 80% cancers detected with 2.4 false positives per case in prospective studies versus 85–90% sensitivity at 1–2 false positives per image in retrospective studies [7].

A. The Role of Neural Networks in CAD

CAD systems usually consist of two distinct subsystems, one designed to detect microcalcifications and one to directly detect masses [8]. A common element in both subsystems is a neural network, used to improve detection and reduce false positive rates. Figure 1 shows a typical CAD system processing flowchart, generalized for either microcalcification or mass detection. The first two stages of the CAD system increase the overall signal-to-noise levels in the image and apply rules/heuristics to define a set of candidate regions-of-interest (ROIs). These stages have adjustable parameters that typically are set to produce a very high sensitivity, usually at a cost of low specificity. The final stage is a statistical model or neural network, whose parameters are found using error-based optimization given a set of training data. The function of this last stage is to reduce false positives (i.e., increase specificity) without significant loss in sensitivity. Neural networks are a particularly important class of statistical models in CAD because they are able to capture complicated, often nonlinear, relationships in high dimensional feature spaces not easily captured by heuristic or rule based algorithms. Several groups have developed neural networks architectures for CAD. Some of these architectures exploit well-known features that might also be used by radiologists [9][10][11], while others utilize more generic feature sets [12][13][14][25]. Both approaches have been shown to be useful for detecting clinically significant mammographic anomalies.

<insert figure 1 here>

B. Exploiting Context in Mammographic Image Analysis

Context can be defined as nearby or surrounding structure that establishes the meaning or identity of an object. In image analysis, contextual information is often used to detect and classify visual objects. For example, detecting a small building in an aerial image can be facilitated by searching along roads, since buildings tend to lie in close proximity to roads. Both human observers and computer vision systems (e.g. [15]) have been developed to exploit contextual relationships in imagery. Likewise, in mammographic image analysis context is exploited by radiologists and mammographers for detecting and identifying breast abnormalities. The clustering of calcifications, their proximity to ductal tissue, the architectural distortion surrounding potential lesions, are all contextual cues used by radiologists and mammographers [16]. Contextual relationships can be integrated into mammographic CAD systems, being made explicit, given known pathology, through incorporation of preset rules and/or feature detectors tuned to capture the context. Alternatively, contextual relationships can be learned from the data, allowing for more complicated and less obvious contextual cues to be uncovered by the pattern recognition system.

C. Overview of Hierarchical Pyramid/Neural Network Architecture

We have developed a pattern recognition architecture that learns contextual relationships between structure in images for detection and classification of objects. Fundamental to the architecture is the multi-scale decomposition of an image, via pyramid transforms [20], and the subsequent integration of multi-scale image features by a hierarchy of neural networks. These fundamental aspects of the architecture led to the name

hierarchical pyramid neural networks (HPNN). Several variants of the HPNN can be defined, dependent upon the direction of processing in the hierarchy. Figure 2 illustrates the general coarse-to-fine and fine-to-coarse architectures. These two architectures detect small or large target object by exploiting coarse-scale (low resolution) or fine-scale (high-resolution) information associated with the target. For example, in the coarse-to-fine HPNN networks operating at low resolution learn contextual features that are passed to networks operating at high resolution and integrated to detect the object of interest (i.e. the contextual inputs condition the probability of target present). For the fine-to-coarse HPNN architecture networks extract detail structure at fine resolutions of the image and then pass this detail information to networks operating at coarser scales (see figure 2B). For many types of objects, information about the fine detail structure is important for discrimination between different classes, i.e., fine resolution structure occurring within the context of the coarse resolution structure is indicative of an object class.

<insert figure 2 here>

We have previously reported on how the HPNN architectures and learning algorithms can improve detection for a general class of image search/detection problems [17][18][19]. For example, we have shown that for the problem of detecting small buildings in aerial imagery, the coarse-to-fine HPNN architecture has higher accuracy than both conventional neural network architectures and standard statistical classification techniques [17]. In this paper we present our results of applying the HPNN framework to two problems in mammographic CAD; detecting microcalcifications and masses in

digital/digitized mammograms. The coarse-to-fine HPNN architecture is well-suited for the microcalcification problem, while the fine-to-coarse HPNN is suited for mass detection. We evaluate the performance and utility of the HPNN framework by considering its effects on reducing false positive rates in a well-characterized CAD system developed by The University of Chicago (UofC). In both cases (microcalcification and mass detection) the HPNN acts as a post-processor of the UofC CAD system.

II. Methods

In this section we describe three critical elements of the HPNN; 1) integrated feature pyramid representation, 2) neural network hierarchy, and 3) the learning algorithm.

A. Integrated Feature Pyramids

Image features are extracted and represented as integrated features pyramids (IFPs) [20]. Multi-scale pyramid transforms are used to construct the IFP, which is the representation that serves as input into the neural network hierarchy. The pyramid transformation for the current set of experiments is based on a general class of filters that measure orientation energy and image intensity gradients.

For the coarse-to-fine IFP, steerable filters [21] are used to compute local oriented gradient information across scale. The steering properties of these filters enable the direct computation of the orientation having maximum energy. Features are constructed which represent, at each pixel location, the maximum energy (energy at orientation θ_{max}), the

energy at the orientation perpendicular to θ_{max} (θ_{max} -90°), and the energy at the diagonal (energy at θ_{max} –45). Figure 3a illustrates the form of the IFP input into the coarse-to-fine network hierarchy.

<insert figure 3 here>

The IFP for mass detection is slightly different from the coarse-to-fine IFP for microcalcification detection (figure 3b). For mass detection, input to the fine-to-coarse neural network hierarchy is an IFP having radial and tangential gradient components at each resolution, relative to the mass center. The features are filtered versions of the image, with filter kernels given by

$$\psi_{q,p}(r,\theta) = \left(\frac{q!}{\pi(q+|p|)!} \right)^{\frac{1}{2}} r^{|p|} e^{\frac{-r^2}{2}} L_q^{|p|}(r^2) e^{ip\phi} \qquad (1)$$

in polar coordinates, with $(q, p) \in \{(0,1),(1,0),(0,2)\}$. These are combinations of derivatives of Gaussians, and can be written as combinations of separable filter kernels and can therefore be computed at relatively low cost. They are also straightforward to steer, being just a multiplication by a complex phase factor. These filters are steered in the radial and tangential directions relative to the mass centers, using the real and imaginary components and their squares and products, as features.[1] Features were

[1] The center coordinates of the masses are generated by earlier stages of the CAD system.

extracted at each level of the Gaussian pyramid representation of the mass ROI, and used as inputs to networks at the same level.

B. Neural Network Hierarchy

The neural networks in the HPNN are multi-layer perceptrons, having one hidden layer with between 4–8 hidden units. The number of hidden units is chosen via cross-validation[22]. All units in a network perform a weighted (w_i) sum of their inputs (x_i), subtracting an offset or threshold (θ) from that sum to get the activation (a)

$$a = \sum_i w_i x_i - \theta \tag{2}$$

The activation is transformed into a unit's output, y, by passing it through a sigmoid function

$$y = \sigma(a) = \frac{1}{1+e^{-a}} \tag{3}$$

Each network in the HPNN hierarchy receives input from the integrated feature pyramid and hidden unit input from networks lower in the hierarchy. Networks are trained either coarse-to-fine or fine-to-coarse, depending on the architecture. In the coarse-to-fine HPNN, the network lowest in the hierarchy is first trained until convergence and then all parameters in this network are held fixed while the next network on the hierarchy is trained. Coarse-to-fine training is possible because the positions of the small objects are well-defined when the resolution is decreased. For the fine-to-coarse HPNN, extended objects do not have a definite location at high resolution.

The entire hierarchy of networks is therefore trained as a single N-layered network (N being a function of the number of layers per network and the number of networks in the hierarchy).[2] Input for both training and testing is raster scanned into each network so that the output of a network at any level is an image. For both HPNN architectures the output of the network is an image representing the probability that an object is present at each x,y position. For the coarse-to-fine architecture each output pixel represents the probability of a point-like object (e.g. a microcalcification), while for the fine-to-coarse architecture each output pixel represents the probability that a large extended object (e.g. a mass) is within that low-resolution pixel.

C. Learning Algorithm

The conventional error function for training a neural network on a binary detection problem is the cross-entropy error function, which is the negative logarithm of the probability that the network produces detection decisions that agree with the targets in the training data. It is given by

$$E = -\sum_i [d_i \log y_i - (1-d_i)\log(1-y_i)] \qquad (4)$$

where $d_i \in \{0,1\}$ is the desired output and y_i is the actual output of the neuron, given by equation 3. For image-based detection, since networks are typically applied across a set of pixels, both y_i and d_i are a function of position; $y_i(x,y), d_i(x,y)$. Thus every position

[2] Error back-propagation through the pyramid reduction operations is straightforward, since this operation is linear.

in an image is either associated with the presence, $d_i(x,y)=1$, or absence, $d_i(x,y)=0$, of a target.

In examining the truth data for the mammographic ROI datasets, we found that radiologists often make small errors in localizing individual microcalcifications and masses. For microcalcifications, these errors appear to be within +/-2 pixels of the correct position. For masses, the positional error also includes the extent of the mass—masses have ill-defined borders that are not easily ground-truthed, even by an expert. If the exact positions of the objects are unknown then the probability of detecting the objects at the correct positions cannot be evaluated and using equation 4 will result in poor performance, as will be illustrated below.

Consider instead the probability of detecting an object of interest when detection is defined as at least one pixel detected within a certain region known to contain the object. For a dataset with a coordinate vector for each object, let $\vec{x}_i$ represent the coordinates of the ith object.[3] Define a region P_i as set of pixel locations for the ith object that incorporate the known magnitude of the uncertainty or positional error in the truth data. A single detection within P_i will represent the detection of the ith object. Denote the output of the network when applied to the input vector derived from the neighborhood of $\vec{x}_i$ to be $y(\vec{x}_i)$. The probability of the network producing at least one detection in P_i is one minus

[3] Note that for analysis of 2D imagery, such as mammograms, $\vec{x}_i=\{x,y\}$. However the formulation can be extended across an arbitrary coordinate space, so we use $\vec{x}_i$ for generality.

the probability of producing no detection in P_i , or $1-\prod_{\vec{x}\in P_i}(1-y(\vec{x}))$. As with cross-entropy, the probability of not detecting an object at a negative position $\vec{x}_i$ is $1-y(\vec{x}_i)$. If we define N as the set of all know negative locations then the new error function becomes;

$$E_{UOP} = -\sum_i \log\left(1-\prod_{\vec{x}\in P_i}(1-y(\vec{x}))\right) - \sum_{\vec{x}_i\in N}\log(1-y(\vec{x})) \qquad (5)$$

We call this the *Uncertain Object Position (UOP)* error function. The first term of equation 5 is the probability of detecting at least one pixel in a positive region while the second term is the probability of no detection in a negative region. The gradient of E_{UOP} with respect to the network weights is

$$\frac{\partial E_{UOP}}{\partial w} = \sum_i \left\{ \frac{\prod_{\vec{x}\in P_i}(1-y(\vec{x}))}{\prod_{\vec{x}\in P_i}(1-y(\vec{x}))-1} \sum_{\vec{x}\in P_i} \frac{\partial y(\vec{x})/\partial w}{(1-y(\vec{x}))} \right\} + \sum_{\vec{x}\in N} \frac{1}{1-y(\vec{x})} \frac{\partial y(\vec{x})}{\partial w} \qquad (6)$$

which is used in an optimization loop for training.[4]

---

[4] a ``weight decay'' regularization term, $r = \frac{\lambda}{2}\sum_i w_i^2$ , is added to the error functions to prevent the networks from becoming ``over-trained''. λ was adjusted to minimize the cross-validation error, computed by dividing the training data into disjoint subsets whose union is the entire set. The network was first trained on all of the training data, and then, starting from this set of weights, the network was retrained on the data with one of the subsets left out. The resulting network was tested on the ``holdout'' subset. This retraining and testing with a holdout set was repeated for each of the subsets, and the average of the errors on the subsets is the cross-validation error, an unbiased estimate of the average error on new data.

As an illustration of the utility of the UOP error function, we compare the detection performance, with a network trained using cross-entropy, for a "toy problem" as shown in figure 4. A 10-by-10 grid of single pixel objects was embedded in a noisy background. Single pixel objects were assigned a pixel value of one, while background pixels had a value of one-half or zero randomly assigned with equal probability. Errors were introduced into the truth data by randomly shifting the truth data within a 3-by-3 pixel neighborhood centered around the object's true position (see figure 5b). A "network" consisting of a single sigmoidal neuron, with activation and transfer functions as in equations 2 and 3, was used to search the image for the objects. At a given location $\vec{x} = \{x, y\}$ the inputs to the network are nine pixel values from a 3-by-3 window in the input image, centered on $\vec{x}$.

<insert figure 4 here>

In figure 4, the truth image shows both the single point truth data and the square 3-by-3 region around these pixels. The images in figure 4d and e are the outputs of the network trained using the cross-entropy error function. The cross-entropy trained network with the output in figure 4d was trained using single point truth data while the network with the output shown in figure 4e was trained using the 3-by-3 region truth data. Figure 4f is the output of the network trained using the UOP error function with positive regions P_i as

shown in figure 4c. As is evident from the figure, the UOP trained network produces qualitatively superior results.

We measured, quantitatively, the detection performance of the networks by computing the sensitivity and false positive rates on the data. For the cross entropy trained networks sensitivity was 90% with a 7.5% pixel false positive rate. For the UOP trained network, sensitivity was 100% with a 0% false positive rate.

III. Results

A. The Experimental Paradigm

We conducted a series of experiments to determine the utility of the HPNN architecture for mammographic CAD. The goal of the first set of experiments was to validate our hierarchical network architecture and learning algorithms for capturing contextual information and to demonstrate improved detection performance, relative to traditional neural network architectures. The second set of experiments focused on a quantitative and rigorous evaluation of the HPNN, in particular evaluation of two architectures for reducing the false positive rate of the state-of-the-art CAD systems developed by UofC. Finally, as a demonstration of clinical utility, we integrated the HPNN with a UofC CAD system and evaluated its performance in a Reader Study.

B. Validation of the network hierarchy architecture

Three neural network architectures were evaluated, each having one hidden layer with 4–8 hidden units.[5] A two level coarse-to-fine IFP was constructed and used as input to the different network architectures. As shown in figure 5, network A consists of a single network processing data from the coarsest resolution of the IFP, network B is a single network receiving input from all levels of the IFP and network C is a 2 level coarse-to-fine HPNN. The networks had activation and transfer functions described previously (equations 2 &3) and were trained using cross-entropy error (equation 4).

We trained the networks on five mammograms. Each mammogram had one or two clusters with approximately 20 microcalcifications per mammogram, for a total of 97. The results given below were measured on five test mammograms with one cluster each, for a total of 95 microcalcifications.

<insert figure 5 here>

Results for the three networks are shown as receiver operating characteristic (ROC) curves [23] in Figure 5. Note the improvement as finer resolution information is added to the network (networks A vs B) and especially the very large improvement when using the hierarchical network architecture (networks A&B vs. C). We considered whether network C was in fact taking advantage of context information by examining the representations developed by various hidden units in the network. Figure 6 shows

outputs of two classes of hidden units. The first class (figure 6 B) appears to represent point-like structure, similar to the structure of an individual microcalcification. The second class of hidden unit (figure 6C) has a different representation. In this case, the unit is selective for long, extended, and oriented structure. When shown to radiologists, they noted that this hidden unit structure appeared correlated with the ductal and vascular anatomy. As mentioned previously, the development of breast cancer is often correlated with these anatomical structures. Results for this experiment suggest that the coarse-to-fine hierarchical neural network is able to automatically extract information that is consistent with known contextual relationships and that this may result in the observed improvement in detection performance.

<insert figure 6 here>

C. Validation of UOP for microcalcification detection

To validate the utility of our UOP error function (equation 5) for mammographic CAD we conducted experiments comparing detection performance with the cross entropy error function (equation 4). We trained and tested a single neuron network to detect microcalcifications, using the dataset described in the previous experiment. Expert radiologists constructed the truth-data, however inspection of the data indicated positional errors of up to 2 pixels. At a given location $\vec{x}$, the inputs to the network were the 25 pixel values in a 5x5 window in the input, centered on $\vec{x}$. We expect that the

[5] Model complexity was controlled for by adding/subtracting hidden units using a cross-

average local brightness is not related to the detection problem. Therefore, to enforce invariance to average local brightness we constrained the weights of the single unit network to sum to one.

Figure 7 shows results for a test mammogram. Note that the network trained using UOP generates fewer false positives than the conventional cross entropy error function. If thresholds are applied to the networks so that 50% of the true positives are detected, the UOP trained network has 50% fewer false positives that the cross entropy network.

<insert figure 7 here>

D. Results on research database: microcalcification detection

Given results for the previous two experiments we next evaluated the performance of an HPNN architecture trained using the UOP error. In the remaining experiments described in this paper we evaluated the performance of the HPNN as a post processor or adjunct for the UofC CAD system.

UofC provided data used for the microcalcification experiments. The first set of data consists of 50 true positive and 86 false positive ROIs. These ROIs are 99-by-99 pixels and digitized at 100μm resolution. A second set of data from the UofC clinical testing database included 47 true positives and 103 false positives, also 99-by-99 and sampled at 100μm resolution.

---

validation error.

We trained a coarse-to-fine HPNN (figure 2A), using UOP error function, to detect individual microcalcifications. Training and testing were done using a jackknife protocol [24], whereby one half of the data (25 TPs and 43 FPs) was used for training and the other half for testing. Results were compiled for five different random splits of the data. For a given ROI, the probability map produced by the network was thresholded at a given value to produce a binary detection map. Region growing was used to count the number of distinct detected regions. The ROI was classified as a positive if the number of regions was greater than or equal to a given cluster criterion.

Table 1 compares ROC results for the HPNN and the shift-invariant artificial neural network (SIANN) network that had been used in the UofC CAD system [25]. Reported are the area under the ROC curve (A_z), the standard deviation of A_z across the subsets of the jackknife (σ_{Az}), the false positive fraction at a true positive fraction of 1.0 (*FPF @TPF=1.0*) and the standard deviation of the FPF across the subsets of the jackknife (σ_{FPF}). A_z and *FPF@TPF*=1.0 represent the averages of the subsets of the jackknife. Note that both networks operate best when the cluster criterion (cc) is set to two. For this case the HPNN has a higher A_z than the SIANN network while also halving the false positive rate. This difference, between the two networks' A_z and *FPF* values, is statistically significant (z-test; p_{Az}=.0018, p_{FPF}=.00001)

<insert table 1 here>

The second set of data was tested using a coarse-to-fine HPNN trained on the first dataset. 150 ROIs taken from a clinical study and classified as positive by the full UofC CAD system for microcalcification detection (including the SIANN neural network) were used to test the HPNN. Though the UofC CAD system classified all 150 ROIs as positive, only 47 were in fact positive while 103 were negatives—this dataset was overpopulated with false positives. We applied the HPNN trained on the entire previous data set to this new set of ROIs. The HPNN was able to reclassify 47/103 negatives as negative, without loss in sensitivity, i.e., no false negatives were introduced.

On examining the negative examples rejected by the coarse-to-fine HPNN, we found that many of these ROIs contained linear, high-contrast structure that would otherwise be false positives for the SIANN network (see figure 8). One possible reason for this is that the coarse-to-fine HPNN also learns context for the false positives. SIANN presumably interprets the "peaks" on the linear structure as calcifications. However because the coarse-to-fine HPNN also integrates information from low resolution it can associate these "peaks" with linear structure at low resolution and thus determine that these peaks are not microcalcifications. This is an interesting difference from our earlier results, in which the networks appeared to learn contextual relationships associated with positive examples—ductal and vascular anatomy. Thus it appears that the HPNN can exploit contextual relationships to both detect true positives and eliminate false positives.

<insert figure 8 here>

E. Results on a research data base: mass detection

The next set of experiments applied a fine-to-coarse HPNN architecture to detect masses in digitized mammograms. Radiologists often distinguish malignant from benign masses based on the detailed shape of the mass border and the presence of spicules along the border [16]. We evaluate the fine-to-coarse HPNN, figure 2B, for its ability to integrate high-resolution information within the context of coarse-scale mass structure.

The experimental paradigm is similar to the microcalcification experiments in that we apply the HPNN as a post-processor to the UofC CAD system for mass detection. The data in our study consists of 72 positive and 100 negative ROIs. The negative ROIs are false-positives of the earlier stages of the CAD system. These are 256-by-256 pixels and are sampled at 200μm resolution.

Results for the fine-to-coarse HPNN system are shown in Table 2. The A_z value on the test set was 0.85. These results show a 51% reduction in false positive rate of the UofC mass detection system without loss in sensitivity.

<insert table 2 here>

F. Results in Clinical evaluation

As a final test of the utility of the HPNN architecture a clinical reader study was conducted to evaluate the performance of the combined HPNN/UoC system for

microcalcification detection.[6] Details of the reader study have been described previously [26]. In this paper we summarize the results.

Table 3 outlines the protocol. Approximately 900 retrospective mammographic cases were collected and read by ten readers. Five readers were considered experts in mammography (spent over 50% of their time reading mammograms) and the other five were general radiologists who were MQSA certified [27]. Films were read in two conditions; film only (unaided) or film + computer results (aided).

<insert table 3 here>

Results of the computer output alone are shown in Table 4. Note that on this new dataset the HPNN continues to reduce the false positive rate of the microcalcification CAD system.

<insert table 4 here>

The clinical utility of the complete system, which includes the CAD systems for mass detection and the HPNN enhanced system for microcalcification detection, is shown in Table 5, comparing reader performance with and without the computer aid. Expert readers showed a statistically significant improvement when using the CAD system, however the improvement was not statistically significant for the general radiologists.

---

[6] In this clinical evaluation only the coarse-to-fine HPNN for microcalcification was integrated

One possible reason is that false positives continue to be an issue, since experts are better than general radiologists at negating or ignoring these false positives. Additional analysis is required to understand the difference between the two groups. However the overall results show that the CAD system, which included the HPNN, can potentially improve performance of mammographic screening, in this case for more experienced radiologists.

<insert table 5 here>

IV. Discussion

In this paper we have demonstrated coarse-to-fine and fine-to-coarse HPNN architectures that learn contextual relationships for detecting microcalcifications and masses in digital/digitized mammograms. Though the architectures are novel, they bear some resemblance to previous network architectures. For example, the fine-to-coarse HPNN is similar to the convolution network proposed by Le Cun, [28], however with a few notable differences. The fine-to-coarse HPNN receives as inputs preset features extracted from the image (in this case radial and tangential gradients) at each resolution, compared to the convolution network, whose inputs are the original pixel values at the highest resolution. Secondly, in the fine-to-coarse HPNN, the inputs to a hidden unit at a particular position are the pixel values at that position in each of the feature images, one pixel value per feature image. Thus the HPNN's hidden units do not learn linear filters, except as linear combinations of the filters used to form the features. Finally the fine-to-coarse HPNN is also trained using the UOP error function, which is not used in the convolution network.

---

with the UofC CAD and evaluated.

The two architectures we have described can be combined into a more general architecture that integrates information both coarse-to-fine and fine-to-coarse. This bi-directional integration, shown in the architecture of figure 9, is attractive in that most objects can be considered to have a "natural scale" -- typically some measure of their size. Classification of the object might be improved through integration of finer and coarser resolution information, relative to this natural scale. Since size can vary within a class of objects, it may be worthwhile to include outputs at more than one level of the HPNN. In this case, the UOP error (Equation 5) needs to be modified to include uncertainty over scale, but this is easily accomplished by changing the product to range over positions at all output levels. We can further generalize the architecture by adding connections between the fine-to-coarse and coarse-to-fine paths, but one must be careful to avoid loops when deciding where these connections should be added. We are currently investigating the application of this generalized HPNN architecture to mass detection.

<insert figure 9 here>

Most of our results were reported relative to the UofC CAD mammographic systems, since they are considered to be well-characterized and state-of-the-art. UofC is continuing to improve upon their systems and our current results are only meant as a comparison to a given standard at a given point in time. An issue in CAD research is the need for the development of appropriate benchmarks for comparing different algorithms.

Several datasets are being developed which might eventually support such comparisons though they have yet to be widely accepted [7].

V. Conclusion

We have presented the application of hierarchical pyramid neural network architectures to two problems in computer-aided diagnosis; the detection of microcalcifications in mammograms and the direct detection of masses in mammograms. In the case of microcalcifications, the coarse-to-fine HPNN architecture successfully discovered large-scale context information that improves the system's performance in detecting small objects. A coarse-to-fine HPNN has been directly integrated with the UofC CAD system for microcalcification detection and the complete system has been tested in clinical reader study. In the case of mass detection, a fine-to-coarse HPNN architecture was used to exploit information from fine resolution detail in order to eliminate false positives. In general, we have found that the HPNN is a useful class of network architecture for exploiting context and integrating information at multiple scales.

Acknowledgments

We would like to thank Drs. Robert Nishikawa and Maryellen Giger of The University of Chicago for fruitful collaborations and discussions as well as providing the mammographic data.

[7] Databases include the Digital Database for Screening Mammography (DDSM), Mammographic Image Analysis Society (MIAS) database, and Lawrence Livermore National Laboratories (LLNL)/University of California at San Francisco (UCSF) database. Information on these and other databases can be obtained from The Digital Mammography Home Page http://www.rose.brandeis.edu/users/mammo/digital.html

Figures

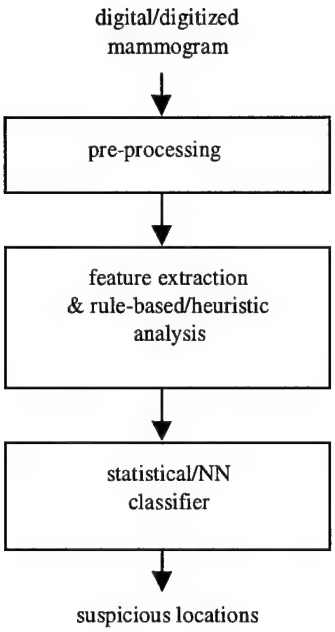

Figure 1 Processing in a CAD system.

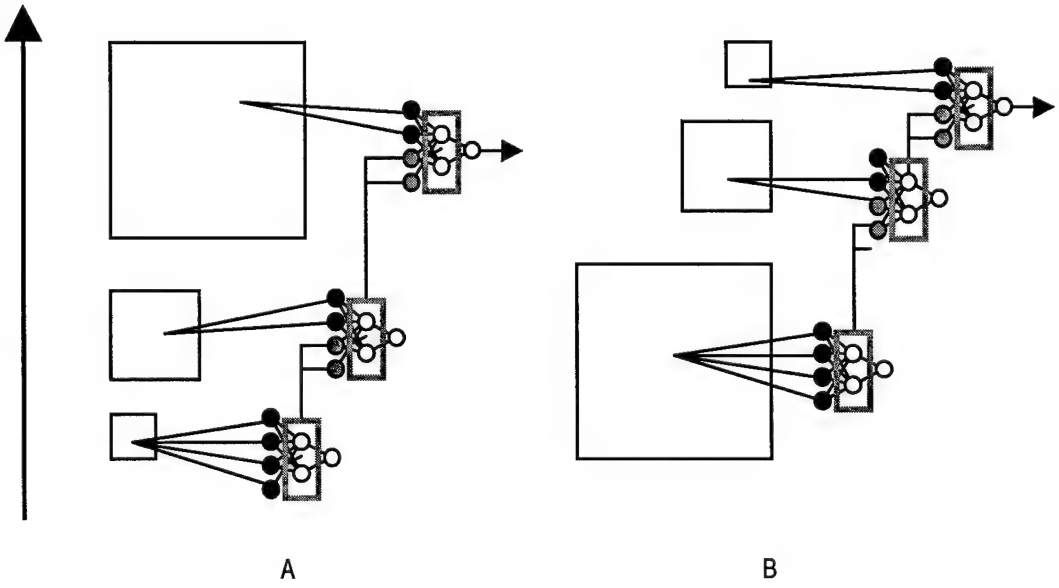

Figure 2: Hierarchical pyramid/neural network architectures. (A) fine-to-coarse and (B) coarse-to-fine. In (A) context is propagated from low to high resolution via the hidden units of low-resolution networks. In (B) small scale detail information is propagated from high to low resolution. In both cases the output of the last integration network is an estimate of the probability that a target is present. Arrow shows direction of information flow.

Figure 3 Integrated Feature Pyramids (IFPs) for A) coarse-to-fine and B) fine-to-coarse HPNN.

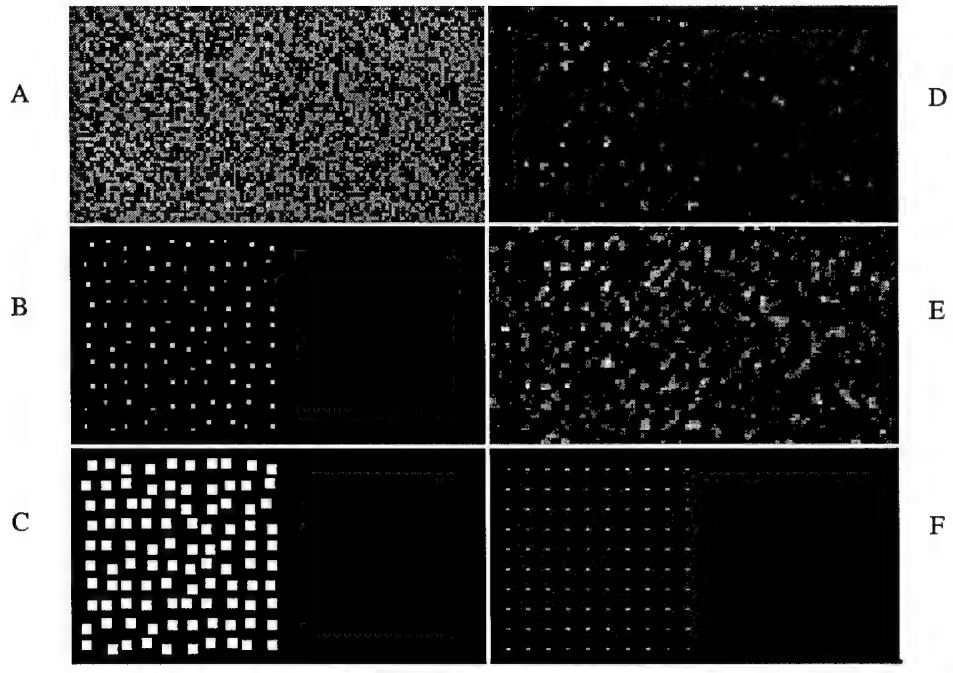

Figure 4: "Toy problem" illustrating performance of UOP error function versus cross-entropy error. (A) Image consisting of 10x10 grid of white dots in a background of random binary noise. (B) Single point truth data with positional error. (C) Truth data created by considering the magnitude of the positional error (+/-1 pixel results in 3X3 regions). (D) Output for network trained using cross-entropy error and truth data in B. (E) Output of network trained using cross-entropy error and truth-data in C. (F) Output of network trained using UOP error and truth data in C.

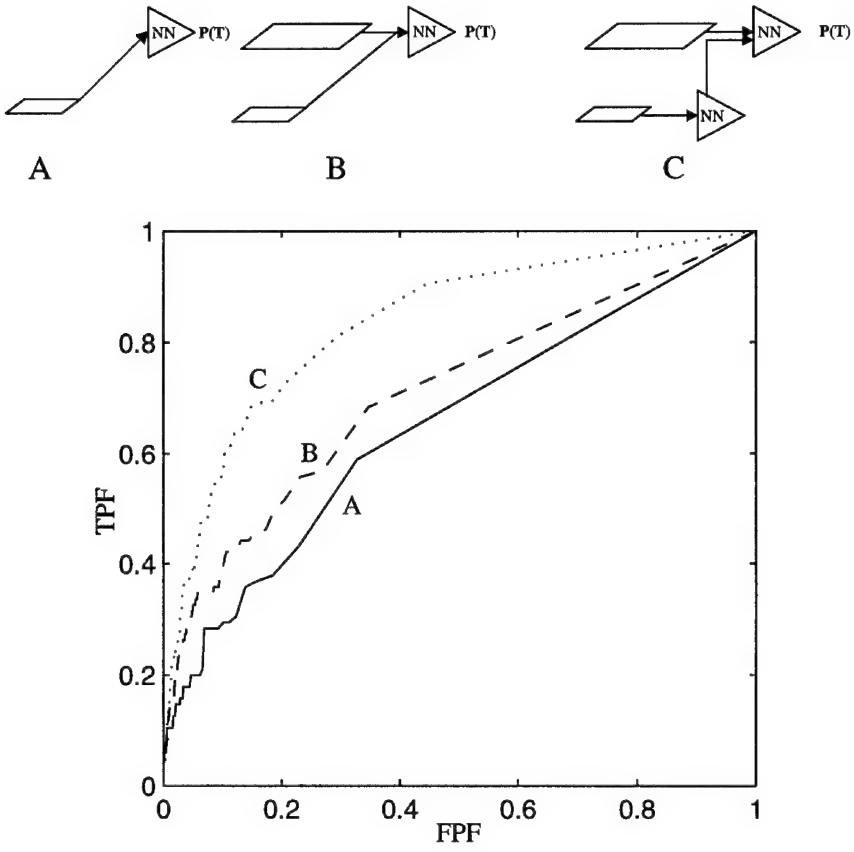

Figure 5: Raw ROC curves for the three networks A, B and C (HPNN).

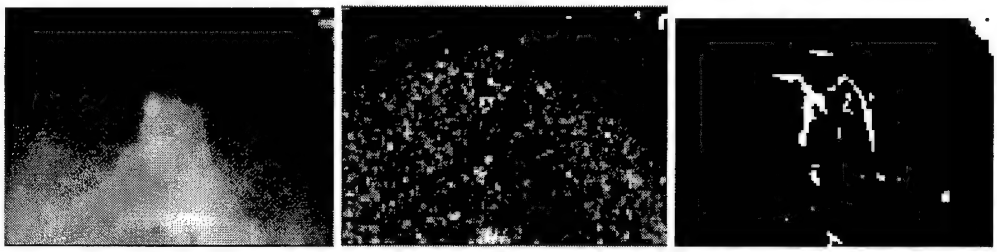

Figure 6 (A) Original mammogram, (B) hidden unit representations for networks operating at high resolutions (C) hidden unit representations for networks operating at low resolutions. Radiologists have noted that some of the structure in C appears to correlate with specific anatomy in the breast (e.g. ducts and/or blood vessels), indicating that these hidden units may represent contextual information.

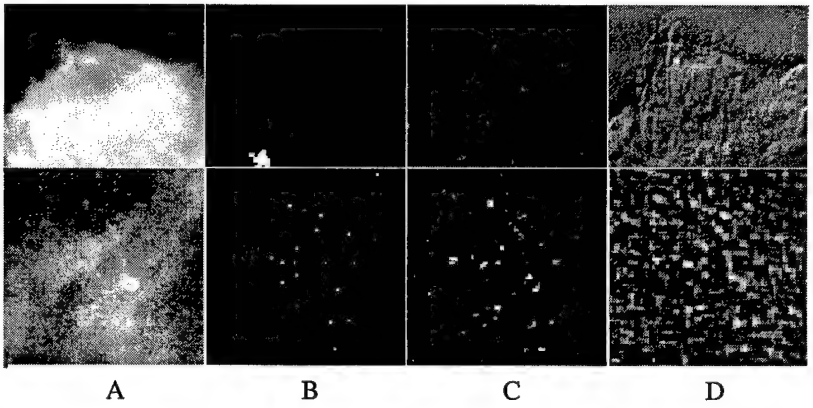

Figure 7 Detecting microcalcifications using UOP error function. The upper row contains reduced resolution images from one full size test mammogram. The lower row shows a region of interest at full resolution. (A) image (B) truth data (C) output of UOP trained network (D) output of cross entropy trained network.

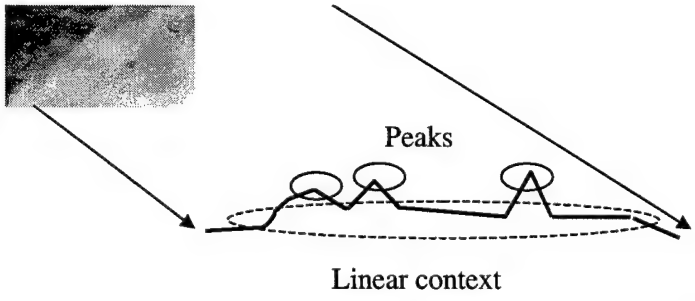

Figure 8 Typical negative ROI that was eliminated by the coarse-to-fine HPNN for microcalcification detection. The HPNN is able to associate the intensity peaks, which in isolation may be interpreted as microcalcfications, with the coarse-scale linear structure in order to classify the ROI as a negative.

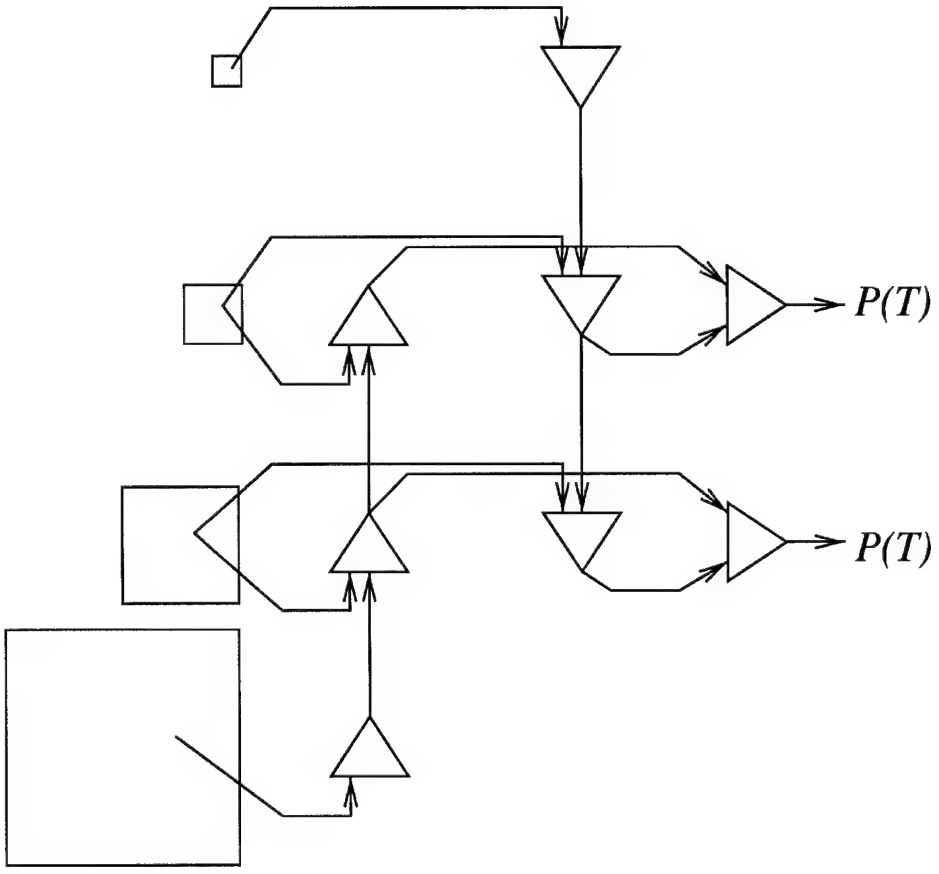

Figure 9 Generalized HPNN architecture. Integration is bi-directional with output networks at the "natural scale" of the object. The natural scale may be known a priori or it can be searched for by optimizing over several output networks (e.g. search for the best one over the two output networks shown above).

Tables

Table 1 Comparison of HPNN and SIANN networks

	HPNN				SIANN			
	A_z	σ_{Az}	FPF TPF=1.0	σ_{FPF}	A_z	σ_{Az}	FPF TPF=1.0	σ_{FPF}
1	.93	.03	.24	.11	.88	.04	.50	.11
2	.94	.02	.21	.11	.91	.02	.43	.10
3	.94	.03	.39	.19	.91	.03	.48	.19
4	.93	.03	.48	.15	.90	.05	.56	.21
5	.93	.03	.51	.06	.88	.05	.68	.21

Table 2 Sensitivity and specificity for fine-to-coarse HPNN for mass detection

Sensitivity	Specificity
100%	51%
95%	57%
90%	67%
80%	79%

Table 3: Summary of Reader Study protocol.

899 cases (4 standard views, original mammograms) • *501 normals (including 10 atypia)* • *199 benign* • *199 malignant (58 DCIS+141 invasive) (22%)*
two reading conditions: • *film only* • *film + computer results* • **films were mounted on alternators** • **computer results were shown on CRT monitors**
standard observer study protocol • *training session randomized reading order, etc.*
10 readers: • *5 specialists (>50% breast imaging)* • *5 general radiologists (MQSA certified)*

Table 4: False positive rates of CAD System

CAD Program	Number false positives per image (at fixed sensitivity)
Mass detection	1.6
Microcalc detection (no HPNN)	1.04
Microcalc detection (with HPNN)	0.88

Table 5: Reader study results using CAD system.

Reader	Specialists		General Radiologists	
	Unaided	Aided	Unaided	Aided
1	0.851	0.878	0.813	0.824
2	0.891	0.911	0.862	0.876
3	0.878	0.898	0.881	0.888
4	0.911	0.914	0.876	0.863
5	0.884	0.903	0.899	0.892
avg	0.883	0.901	0.866	0.869
p value	0.01		0.19	

References

[1] K. Doi, M.L. Giger, R.M. Nishikawa, K. Hoffmann, H. MacMahon, R.A. Schmidt, and K.G. Chua, "Digital radiography: A useful clinical tool for computer-aided diagnosis by quantitative analysis of radiographic images," *Acta Radiologica*, vol 34, pp. 426-439, 1993.

[2] R.E. Bird , "Professional quality assurance for mammography screening programs," *Radiology*, vol 177, pp.8-10, 1990.

[3] C.E. Metz, J.H. Shen., "Gains in accuracy from replicated readings of diagnostic images: prediction and assessment in terms of ROC analysis," *Medical Decision Making*, vol 12, pp. 60-75, 1992.

[4] E.L. Thurfjell, K.A. Lernevall, and A.S. Taube, "Benefit of independent double reading in a population-based mammography screening program," *Radiology*, vol 191, pp. 241-244, 1994.

[5] R2 Technology Pre-market approval (PMA) of the M1000 Image Checker, FDA application #P970058, approved June 26, 1998.

[6] L.J. Burhenne, S.A. Wood, C.J. D'Orsi,, S.A. Feig, D.B. Kopans, K.F. O'Shaughnessy, E.A. Sickles, L. Tabar, C.J. Vyborny, and R.A. Castellino, "Potential Contribution of Computer-aided Detection to the Sensitivity of Screening Mammography," *Radiology*, vol 215, pp.554–562, 2000.

[7] R.M. Nishikawa, R.A. Schmidt, R.B. Osnis, M.L. Giger, K. Doi, and D.E. Wolverton, "Two-year Evaluation of a Prototype Clinical Mammographic Workstation for Computer-aided Diagnosis," *Radiology*, 201 (P), 256, 1996.

[8] M.L. Giger, Z. Huo, M.A. Kupinski, and C.J. Vyborny, "Computer-aided Diagnosis in Mammography," In *Handbook of Medical Imaging; Volume 2. Medical Image Processing and Analysis*, M. Sonka and J.M. Fitzpatrick, JM, eds. SPIE Press pp. 917-986, 2000.

[9] Z. Huo, M.L. Giger, C.J. Vyborny, D.E. Wolverton, R.A. Schmidt, and K. Doi, "Automated computerized classification of malignant and benign mass lesions on digital mammograms," *Academic Radiology*, vol. 5, pp. 155-168, 1998.

[10] C.E. Floyd, J.Y. Lo, A.J. Yun, D.C. Sullivan, P.J. Kornguth, "Prediction of breast cancer malignancy using an artificial neural network," *Cancer,* vol 74, pp.2944-2948, 1994.

[11] Y. Jiang, R.M. Nishikawa, D.E. Wolverton, C.E. Metz, M. L. Giger, R.A. Schmidt, and K. Doi, "Automated feature analysis and classification of malignant and benign microcalcifications," *Radiology*, vol 198, pp. 671-678, 1996.

[12] H.P. Chan, B. Sahiner, K. L. Lam, N. Petrick, M. A. Helvie, M.M. Goodsitt, and D.D. Adler, "Computerized analysis of mammographic microcalcifications in morphological and feature spaces," *Medical Physics*, vol. 25, pp. 2007-2019, 1998.

[13] J.Y. Lo, J. Kim, J.A. Baker, and C.E. Floyd, "Computer-aided diagnosis of mammography using an artificial neural network: Predicting the invasiveness of breast cancers from image features," In *Medical Imaging 1996: Image Processing*, M.H. Loew, ed., Proc. SPIE vol 2710, pp.725-732, 1996.

[14] S.C. Lo, H.P. Chan, J.S. Lin, H. Li, M.T. Freedman, and S.K. Mun, "Artificial Convolution Neural Network for Medical Image Pattern Recognition," *Neural Networks*, vol. 8(7/8), pp. 1201-1214, 1995.

[15] *IEEE Workshop on Context-based Vision*, J. Mundy and T. Strat, organizers, 1995.

[16] D.B. Kopans, *Breast Imaging*, J.B. Lippincott Company, Philadelphia, 1989.

[17] P. Sajda, C.D. Spence, S. Hsu and J.C. Pearson, "Integrating Neural Networks with Image Pyramids to Learn Target Context," *Neural Networks,* vol 8(7/8), pp.1143–1152, 1995.

[18] C. Spence, P. Sajda, S. Hsu and J. Pearson, "Neural Network/Pyramid Architectures that Learn Target Context," *DARPA Image Understanding Workshop*, pp.853-862, 1994.

[19] C. Spence. Supervised learning of detection and classification tasks with uncertain training data. In *ARPA Image Understanding Workshop,* pp. 1395-1402, 1996.

[20] P.J. Burt, " Smart sensing within a pyramid vision machine," *Proceedings IEEE,* vol 76(8), pp. 1006–1015, 1988. Also in *Neuro-Vision Systems,* Gupta and Knopf, eds., 1994.

[21] W.T. Freeman and E.H. Adelson. "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-13(9):891–906, 1991.

[22] C. Bishop. *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[23] C. Metz, "Current problems in ROC analysis," In *Proceedings of the Chest Imaging Conference*, pp 315–33, Madison, WI, November 1988.

[24]K. Fukunaga, *Introduction to Statistical Pattern recognition, 2nd edition,* Academic Press Inc., New York, 1990.

[25] W. Zhang, K. Doi, M. L. Giger, Y. Wu, R. M. Nishikawa, and R. Schmidt, "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network," *Medical Physics,* vol 21(4), pp. 517–524, 1994.

[26] R.M. Nishikawa, C. Gatsonis, M.D. Schnall, M.L. Giger, P. Sajda, and M. Chen, "Large Scale Observer Study to Measure the Benefits of Computer-aided Detection to Screening Mammography," *Radiology*, 213 (P), 150, 1999.

[27] Mammography Quality Standards Act (MQSA) of 1992.

[28] Y. Le Cun, B. Boser, J.S. Denker, and D. Henderson, "Handwritten digit recognition with a back-propoagation network, " In *Advances in Neural Information Processing Systems 2*, D.S. Touretzky, editor, pp. 396-404, 1990.